Oracle® Identity Governance

Configuring a Flat File-Based Application

12c (12.2.1.3.0)

F14155-02

March 2023

Oracle Identity Governance Configuring a Flat File-Based Application, 12c (12.2.1.3.0)

F14155-02

Primary Author: Alankrita Prakash

Contributors: Raghunath Edhara, Azra Shakeel

Contents

Preface

What's New in This Guide?

1 About the Flat File Connector

6 Using the Flat File Connector

7 Extending the Functionality of the Flat File Connector

8 Upgrading the Flat File Connector

9 Troubleshooting the Connector

List of Figures

List of Tables

Preface

This guide describes the connector that is used to onboard applications for flat files, exported from various enterprise target systems, to Oracle Identity Governance.

Audience

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

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For information about installing and using Oracle Identity 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 Manager 11.1.2.3, visit the following Oracle Help Center page:

`http://docs.oracle.com/cd/E52734_01/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.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

What's New in This Guide?

These are the updates made to the software and documentation for release 12.2.1.3.0.

The updates discussed in this chapter are divided into the following categories:

- Software Updates

  This section describes updates made to the connector software.

- Documentation-Specific Updates

  This section describes major changes made to this guide. These changes are not related to software updates.

Software Updates

These are the updates made to the connector software.

Software Updates in Release 12.2.1.3.0

The following is the software update in release 12.2.1.3.0:

Support for Onboarding Applications Using the Connector

From this release onward, the connector bundle includes application onboarding templates required for performing connector operations on a flat file. This helps in quicker onboarding of flat file-based applications into Oracle Identity Governance by using an intuitive UI.

Documentation-Specific Updates

These are the updates made to the connector documentation.

Documentation-Specific Updates in Release 12.2.1.3.0

This is the first release of this connector. Therefore, there are no documentation-specific updates in this release.

1

About the Flat File Connector

The Flat File connector integrates Oracle Identity Governance with files of formats such as CSV and XML.

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

1.1 Introduction to the Connector

Oracle Identity Governance is a centralized identity management solution that provides self service, compliance, provisioning and password management services for applications residing on-premises or on the Cloud. Oracle Identity Governance connectors are used to integrate Oracle identity Governance with the external identity-aware applications.

The Flat File connector lets you onboard flat file-based applications in Oracle Identity Governance.

> **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).

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.

Enterprise applications generally support the export of users in the form of a file. Some widely used file formats are CSV, LDIF, and XML. The connector consumes the information in a flat file, thereby enabling the import of this data as Oracle Identity Governance user accounts or entitlements. You can use the flat file connector in a number of situations for offline data loading or when a predefined connector is not available.

By default, this connector supports processing of flat files in the CSV format. If you want to use this connector to process flat files in formats other than CSV, then you must create a custom parser and integrate it with the connector.

1.2 Certified Components

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

Table 1-1 Certified Components

Item	Requirement for AOB Application	Requirement for CI-Based Connector
Oracle Identity Governance or Oracle Identity Manager	Oracle Identity Governance 12c PS4 (12.2.1.4.0)	You can use one of the following releases: • Oracle Identity Governance 12c PS3 (12.2.1.3.0) • Oracle Identity Manager 11*g* Release 2 PS3 (11.1.2.3.0) and later
Target System	Any enterprise system that can export users, accounts, or entitlements to a flat file.	Any enterprise system that can export users, accounts, or entitlements to a flat file.
Connector Server	11.1.2.1.0	11.1.1.5.0 or later **Note:** You can download the necessary Java Connector Server from the Oracle Technology Network web page.
Connector Server JDK	JDK 1.8 or later	JDK 1.8 or later
Flat File format	CSV **Note:** Formats other than CSV are supported through the use of custom parsers.	CSV **Note:** Formats other than CSV are supported through the use of custom parsers.

1.3 Certified Languages

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

Resource bundles are not part of the connector installation media as the resource bundle entries vary depending on the flat file being used. You can localize field labels in UI forms as described in Localizing Field Labels in UI Forms.

1.4 Usage Recommendation

These are the recommendations for the Flat File connector versions that you can deploy and use depending on the Oracle Identity Governance or Oracle Identity Manager version that you are using.

- If you are using Oracle Identity Governance 12c PS4 (12.2.1.4.0), then use the latest 12.2.1.*x* version of this connector. Deploy it using the **Applications** option on the Manage tab of Identity Self Service.
- If you are using Oracle Identity Governance 12c PS3 (12.2.1.3.0) or Oracle Identity Manager 11*g* Release 2 PS3 (11.1.2.3.0) and later, then use the 11.1.1.*x* version of the Flat File connector. If you want to use the 12.2.1.*x* version of this connector, then you can install and use it only in the CI-based mode. If you want to use the AOB application, then you must upgrade to Oracle Identity Governance release 12.2.1.4.0.

> **Note:**
>
> If you are using the latest 12.2.1.*x* version of the Flat File connector in the CI-based mode, then see *Oracle Identity Manager Connector Guide for Flat File*, Release 11.1.1 for complete details on connector deployment, usage, and customization.

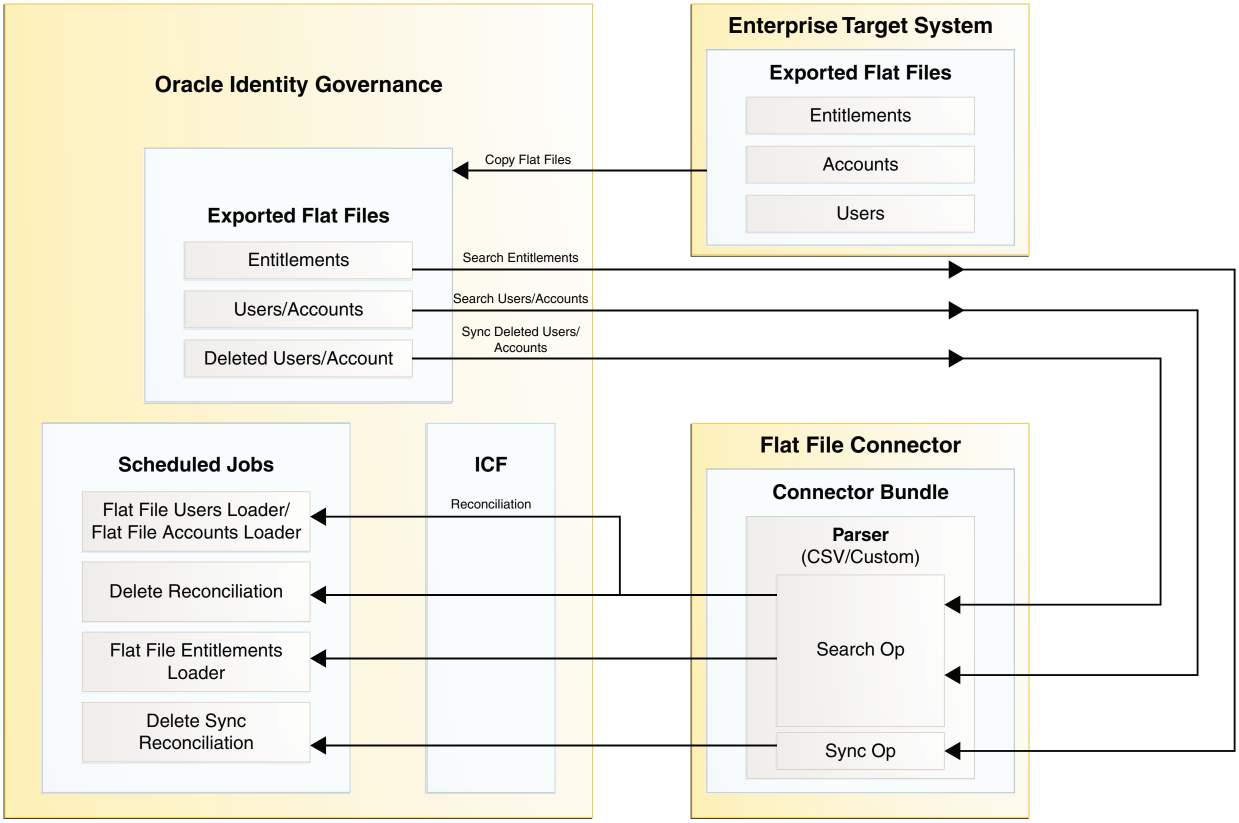
1.5 Connector Architecture

The Flat File connector is a generic solution to retrieve records from flat files that are exported from various enterprise target systems. This connector is implemented using the Identity Connector Framework (ICF) component.

These flat files can be of various formats such as CSV, LDIF, XML, and so on. The connector focuses only on the reconciliation of records from a flat file. The connector installation package contains scheduled jobs that you can use to load users, accounts, and entitlements from a flat file into an existing resource in Oracle Identity Governance.

Figure 1-1 shows the connector integrating the flat files exported from an enterprise target system with Oracle Identity Governance.

Figure 1-1 Architecture of the Connector



You must place the flat files exported from the enterprise target system in a directory that is accessible from Oracle Identity Governance. If you are using a Connector Server, then place the exported flat files on the computer hosting the Connector Server. The connector sorts the exported flat files within the directory in an alphanumeric manner, and then processes each file based on this order.

The location of the directory containing the flat file is specified in the attributes of a scheduled job. When a scheduled job is run, it calls the connector's search implementation, which in turn returns the connector objects to Oracle Identity Governance.

1.6 Use Cases Supported by the Connector

These are the scenarios in which you can use the connector.

- Reconciliation of Records
- Disconnected Resource
- Connected Resource

1.6.1 Reconciliation of Records

Reconciling records from a flat file exported from an enterprise target system involves loading data from a flat file into Oracle Identity Manager.

You can perform the following operations in this scenario:

- Reconciliation
- Certification

Here, the Flat File connector can be used to perform reconciliation runs.

The following example shows how the Flat File connector can be used to load data from a flat file into Oracle Identity Governance to perform certification tasks.

Suppose John works as a Compliance Administrator in ACME Corporation. He uses Oracle Identity Governance to define roles, automate certification processes, and generate business structure reports for auditing. He has a list of users in his enterprise and their entitlements in the form of a CSV file, and he wants to import this data into Oracle Identity Governance, to use this data purely for certification purposes. He needs to create resource objects and forms for all the users, and import the data into these tables.

In the preceding example, by using the flat file connector, John can load accounts from a flat file into a Flat File Resource. He can run the corresponding reconciliation jobs of the flat file to import data from the CSV file into Oracle Identity Governance.

1.6.2 Disconnected Resource

Disconnected resources are targets for which a predefined connector does not exist. Therefore, the provisioning fulfillment for disconnected resources is not automated, but manual.

You can perform the following operations in this scenario:

- Request
- Manual fulfillment or provisioning
- Reconciliation
- Certification

Here, the Flat File connector can be used to perform reconciliation runs and provisioning operations.

The following example shows how the Flat File connector can be used to load data from a flat file into Oracle Identity Governance for disconnected resources.

Suppose Smith is the chief librarian in the University of Utopia. His responsibilities include providing library access cards to the students of the university. He has a file with the list of students who already have library cards. He wants to transfer this list to Oracle Identity Governance after which he can automate the library transactions for existing members.

In the preceding example, as library cards are modeled as a disconnected resource in Oracle Identity Governance, he can create an application for the disconnected resource, and then load accounts from a flat file into a Library Card Resource using the corresponding reconciliation jobs. By defining a disconnected resource through Oracle Identity Governance, Smith can start reconciling users from the flat file and link them to the desired disconnected resource.

1.6.3 Connected Resource

Connected resources are targets for which a predefined connector is available, for example, Microsoft Active Directory.

You can perform the following operations in this scenario:

- Request
- Automatic fulfillment or provisioning
- Reconciliation
- Certification

Here, the Flat File connector can be used only to perform reconciliation runs.

The following example shows how the Flat File connector can be used to load data from a flat file into Oracle Identity Governance, although a predefined connector is available.

Suppose Jane works as a Network Administrator at Example Multinational Inc. In Example Multinational Inc., she performs identity and access management tasks on users within the organization. One of Jane's responsibilities is to create and maintain users in Oracle Identity Governance, and to provision these users with resources. At Example Multinational Inc., all the employee details are maintained in the Microsoft Active Directory target system. Jane wants to reconcile about 100,000 user records from the target system to her Oracle Identity Governance instance, as soon as possible. As the AD Server is planned for a maintenance shutdown, she is looking for a means for offline loading of all the user data which has been exported in the form of an LDIF file. Given the time and network constraints, Jane needs a solution for the initial on-boarding of the users into Oracle Identity Governance.

In the preceding example, performing an initial reconciliation or full reconciliation, is a performance and time-intensive operation. Using the Microsoft Active Directory User Management connector to perform the reconciliation operation requires the connection between the target system and Oracle Identity Governance to remain active. In other words, offline loading of users cannot be performed. In this scenario, a native flat file dump from the target system can be used by the Flat File connector to quickly reconcile the users into Oracle Identity Governance.

1.7 Features of the Connector

The features of the connector include support for custom parsers, fault handling, archival, connector server, transformation and validation of account data, full, incremental, limited, and batched reconciliation, and so on.

- Support for Both Target Resource and Trusted Source Reconciliation
- Full and Incremental Reconciliation
- Limited Reconciliation
- Support for Disconnected Resources
- Support for Archival
- Support for Custom Parsers
- Support for Reconciling Complex Multivalued Data
- Support for Delimiters
- Support for Comment Characters
- Support for Fault Handling
- Support for Preprocess and Postprocess Handlers

- Transformation and Validation of Account Data
- Support for the Connector Server

1.7.1 Support for Both Target Resource and Trusted Source Reconciliation

You can configure the exported flat file as a Target application or an Authoritative application for reconciliation of records into Oracle Identity Governance.

See Providing Settings Information for a Disconnected Resource and Providing Settings Information for a Connected Resource for more information about the reconciliation jobs that are created when you create the application and their details.

1.7.2 Full and Incremental Reconciliation

After you create the application, you can perform full reconciliation to load all existing user data from the flat file to Oracle Identity Governance.

Any new files that are added after the first full reconciliation run are considered as a source of incremental data. Alternatively, incremental reconciliation can also be performed by explicitly providing the incremental data alone.

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

1.7.3 Limited Reconciliation

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

See Performing Limited Reconciliation for more information.

1.7.4 Support for Disconnected Resources

The connector provides support for disconnected resources by generating all artifacts associated with disconnected resources.

In addition, it generates process definitions associated with the default SOA composites that are required for performing manual provisioning. This eliminates the need to manually create disconnected resources and mappings between fields in Oracle Identity Governance and corresponding target system attributes.

To configure your flat file as a disconnected resource, see Creating an Application for a Disconnected Resource By Using the Flat File Connector

1.7.5 Support for Archival

The connector supports archival of the processed flat files.

You can specify the archive directory location in the Archive Directory parameter while configuring the reconciliation jobs, and the connector moves the files from the source directory to the specified location, once all files are processed.

If you do not specify a value for this parameter, then the connector creates a directory named "archived" within the directory containing the flat file, and saves the processed files in this location.

The Oracle Identity Governance Administrator must have read and write permissions on the Archive directory location.

The connector saves the processed flat file in the following format:

FILENAME`_dd-MM-yyyy_HH-mm-ss.zip`

In this format:

- *FILENAME* is the name of the flat file being archived. If the directory with the flat file that is being processed contains more than one flat file, then *FILENAME* is the name of the first flat file from the alpha-numerically sorted list of flat files in the directory.
- `dd-MM-yyyy_HH-mm-ss` is the date and time at which the flat file was archived.

For example, if the flat file has been exported from an enterprise target system, then the filename is saved in the following format:

```
acmeusers_29-08-2013_22-44-12.zip
```

When the archive location is specified, the connector moves all the files from the source directory irrespective of whether the file processing was successful or not. In case of errors, the connector writes the failed records to a separate file and this file is saved in the "failed" directory under the Flat File directory.

See Support for Fault Handling for more information about files in the "failed" directory.

1.7.6 Support for Custom Parsers

By default, the connector supports processing of flat files exported in the CSV format. To support the processing of flat files exported in formats other than CSV, you must create a custom parser and integrate it with the connector.

By default, the connector installation media contains the CSVParser.

See Configuring Custom Parsers for more information about custom parsers.

1.7.7 Support for Reconciling Complex Multivalued Data

The connector supports the reconciliation of complex multivalued data in the form of child forms containing single and multiple fields.

The child form data must be in the same file as the parent form data. The child form values are separated by customizable delimiters.

For example, in CSV files, every line in the flat file represents a single record which includes the parent and the child form data.

See Configuring the Flat File for Reconciliation of Complex Multivalued Data for more information.

1.7.8 Support for Delimiters

The connector supports the use of single character delimiters, which are used to separate values in a record.

By default, the connector supports comma (,) as a fieldDelimiter, semicolon (;) as a multiValueDelimiter, and number sign (#) as a subFieldDelimiter. If the exported flat file uses other characters as delimiters, then specify them as the values for the fieldDelimiter, multiValueDelimiter, and subFieldDelimiter parameters of the Advanced Settings section.

You must specify the Space or tab characters as `space` or `tab` respectively. Other multibyte characters (characters in different locale) can be directly entered in the Advanced Settings section of the respective locale.

> **Note:**
>
> The connector does not support multicharacter delimiters. For example, the use of characters `$#` together as a delimiter is not supported.

In the following sample multivalued data, the data has been presented in the following format, separated by delimiters:

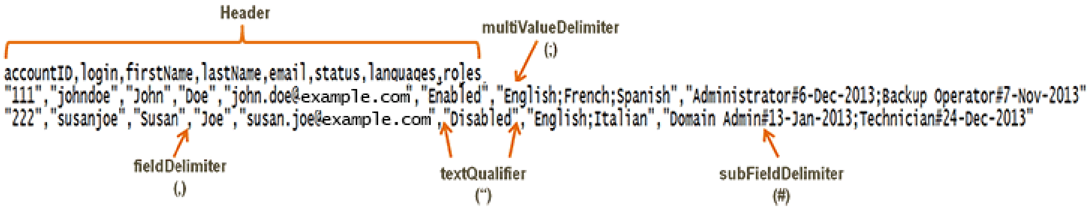
AccountID,FirstName,LastName,Email,Languages,Roles

```
"111","John","Doe","john.doe@example.com","English;French;Spanish","Administrator#6-
Dec-2013;Backup Operator#7-Nov-2013"
```

Here, comma (,) is a fieldDelimiter, semicolon (;) is a multiValueDelimiter, and number sign (#) is a subFieldDelimiter.

Figure 1-2 shows sample multivalued data separated by delimiters.

Figure 1-2 Sample Multivalued Data Separated by Delimiters



```
accountID,login,firstName,lastName,email,status,languages,roles
"111","johndoe","John","Doe","john.doe@example.com","Enabled","English;French;Spanish","Administrator#6-Dec-2013;Backup Operator#7-Nov-2013"
"222","susanjoe","Susan","Joe","susan.joe@example.com","Disabled","English;Italian","Domain Admin#13-Jan-2013;Technician#24-Dec-2013"
```

See Providing Basic Information for a Connected Resource or Providing Basic Information for a Disconnected Resource for more information about the fieldDelimiter, multiValueDelimiter, and subFieldDelimiter parameters.

1.7.9 Support for Comment Characters

You can configure the connector to ignore the processing of lines that begin with certain characters such as #,$, and so on.

These configurable characters are considered as comment characters, and sentences beginning with such characters are considered as comments. The connector implementation skips any lines that start with the configured comment character.

You can configure this by adding an attribute named commentCharacter in the Advanced Settings section of your flat file application as described in Configuring the Connector to Ignore Comment Characters.

1.7.10 Support for Fault Handling

The connector logs record-level errors in a separate file while parsing the flat file. This log file is saved in a directory named "failed" that the connector creates, within the flat file directory.

The connector saves the processed flat file in the following format:

FILENAME`_dd-MM-yyyy_HH-mm-ss.`*EXT*

In this format:

- *FILENAME* is the name of the flat file being archived.
- `dd-MM-yyyy_HH-mm-ss` is the date and time at which the connector started processing the file.
- *EXT* is the extension of the file.

For example, the filename will be saved in the following format:

```
acmeusers_29-08-2013_22-44-12.csv
```

The error file contains all those records that were not processed due to validation or data errors. The connector also appends the reason for failure as a separate attribute in the error file for future reference. Since the error file contains the existing attributes of the failed record, you can modify the same file to fix the data errors and load it back using the connector to reconcile the failed records alone. The Oracle Identity Governance Administrator must have read and write permissions on the Flat File directory and Archive directory locations.

1.7.11 Support for Preprocess and Postprocess Handlers

Preprocess and postprocess tasks can be run both before and after the reconciliation of accounts respectively.

You can use these tasks to perform any job on the flat file directory, like zipping and unzipping files, encryption and decryption of the complete file dumps or specific fields in the files, virus scan of the files, or any other tasks limited only by the implementation of these tasks.

See Configuring Preprocess and Postprocess Tasks for more information.

1.7.12 Transformation and Validation of Account Data

You can configure transformation and validation of account data that is brought into or sent from Oracle Identity Governance during reconciliation and provisioning operations by writing Groovy scripts while creating your application.

For more information, see Validation and Transformation of Provisioning and Reconciliation Attributes in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

1.7.13 Support for the Connector Server

Connector Server is one of the features provided by ICF. By using one or more connector servers, the connector architecture permits your application to communicate with externally deployed bundles.

A Java connector server is useful when you do not wish to execute a Java connector bundle in the same VM as your application. It can be beneficial to run a Java connector on a different host for performance improvements.

For information about installing, configuring, and running the Connector Server, and then installing the connector in a Connector Server, see Using an Identity Connector Server in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*.

2 Prerequisites for Creating and Configuration Applications for Disconnected and Connected Resources

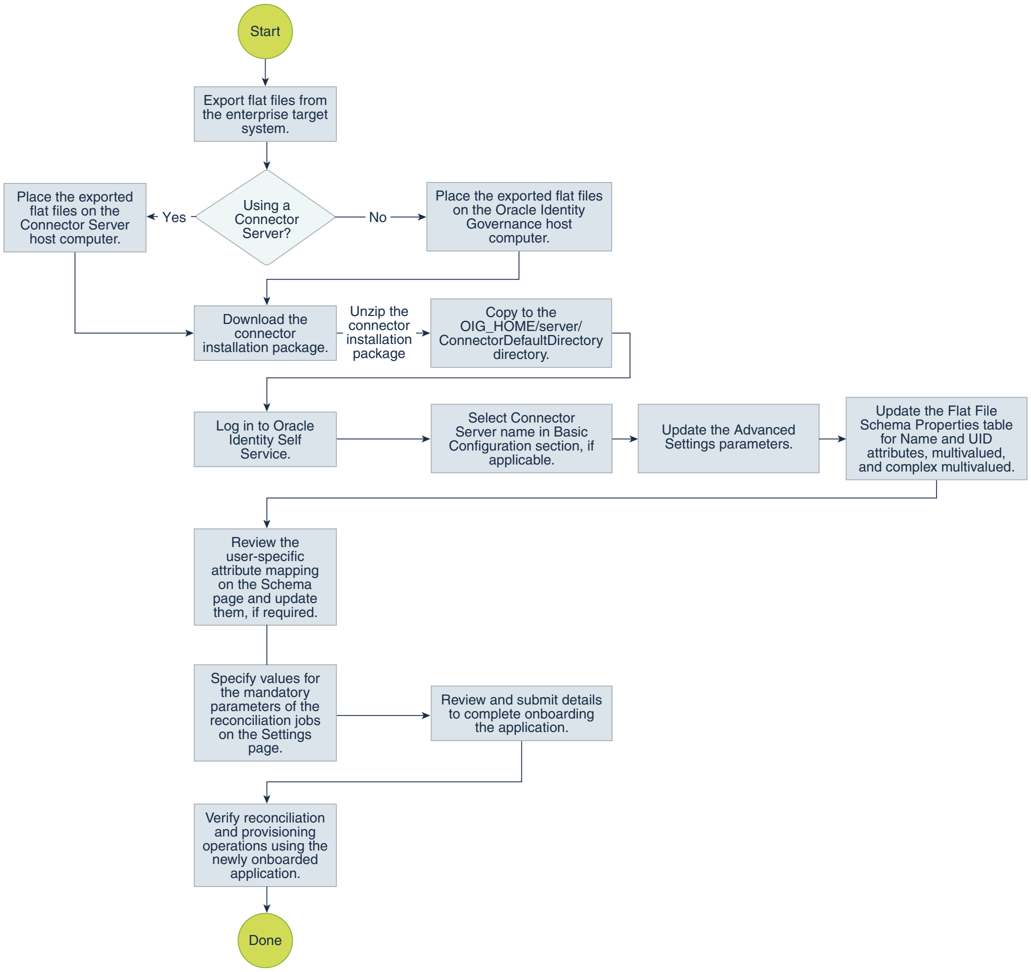
Learn about the tasks that you must complete before you create an application for a disconnected resource and before you configure flat files for a connected application.

2.1 Process Flow for Creating an Application for a Disconnected Resource

From Oracle Identity Governance release 12.2.1.4.0 onward, connector creation and 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 for a disconnected resource by using the connector installation package.

Figure 2-1 Overall Flow of the Process for Creating an Application for a Disconnected Resource

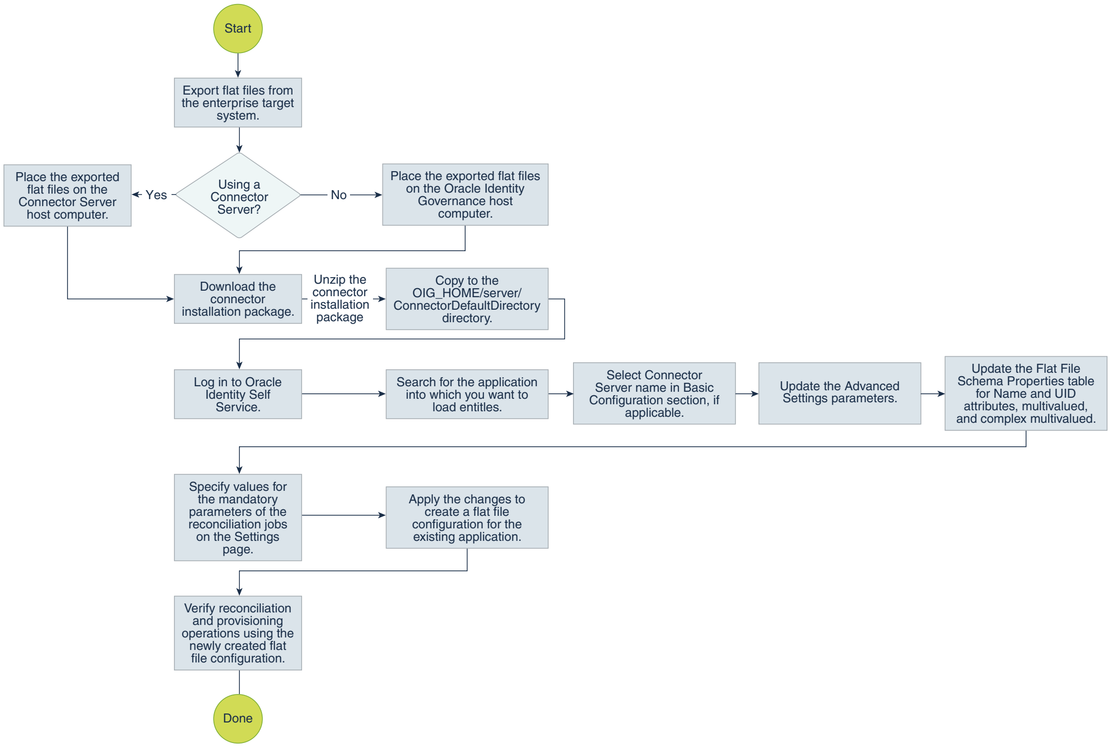

Start
Export flat files from the enterprise target system.
Using a Connector Server?
Yes
Place the exported flat files on the Connector Server host computer.
No
Place the exported flat files on the Oracle Identity Governance host computer.
Download the connector installation package.
Unzip the connector installation package
Copy to the OIG_HOME/server/ ConnectorDefaultDirectory directory.
Log in to Oracle Identity Self Service.
Select Connector Server name in Basic Configuration section, if applicable.
Update the Advanced Settings parameters.
Update the Flat File Schema Properties table for Name and UID attributes, multivalued, and complex multivalued.
Review the user-specific attribute mapping on the Schema page and update them, if required.
Specify values for the mandatory parameters of the reconciliation jobs on the Settings page.
Review and submit details to complete onboarding the application.
Verify reconciliation and provisioning operations using the newly onboarded application.
Done

2.2 Process Flow for Configuring an Application for a Connected Resource

From Oracle Identity Governance release 12.2.1.4.0 onward, connector creation and deployment is handled using the application onboarding capability of Identity Self Service.

Figure 2-2 is a flowchart depicting high-level steps for configuring a flat file application in Oracle Identity Governance for a connected resource by using the connector installation package.

Figure 2-2 Overall Flow of the Process for Configuring an Application for a Connected Resource


Start
Export flat files from the enterprise target system.
Using a Connector Server?
Yes
Place the exported flat files on the Connector Server host computer.
No
Place the exported flat files on the Oracle Identity Governance host computer.
Download the connector installation package.
Unzip the connector installation package
Copy to the OIG_HOME/server/ ConnectorDefaultDirectory directory.
Log in to Oracle Identity Self Service.
Search for the application into which you want to load entitles.
Select Connector Server name in Basic Configuration section, if applicable.
Update the Advanced Settings parameters.
Update the Flat File Schema Properties table for Name and UID attributes, multivalued, and complex multivalued.
Specify values for the mandatory parameters of the reconciliation jobs on the Settings page.
Apply the changes to create a flat file configuration for the existing application.
Verify reconciliation and provisioning operations using the newly created flat file configuration.
Done

2.3 Installing the Prerequisite Software

To configure flat files and load entities into a resource for which a predefined connector is available, ensure that you have created the application for the predefined connector in Oracle Identity Governance.

> **Note:**
>
> Creating an application for the predefined connector is a prerequisite only if you are configuring flat files for a connected resource.

For a list of connectors that you can install and use, visit the Oracle Identity Governance Connectors Documentation website at https://docs.oracle.com/middleware/oig-connectors-12213/index.html

2.4 Exporting Flat Files

From your enterprise target system, you must export the flat file that contains the records that you want to reconcile into Oracle Identity Governance.

Ensure that you export all the flat files for user management and entitlement management. For example, export the flat file containing user account data and export one flat file each for every entitlement (groups, or roles, or departments) that you may have. Ensure that you place all your exported flat files at a location that is accessible from the computer hosting Oracle Identity Governance.

If you are using a Connector Server, then all the exported flat files must be present on the computer on which the Connector Server is installed.

2.5 Configuring the Flat File for Reconciliation of Complex Multivalued Data

The Flat File connector supports the reconciliation of complex multivalued data in the form of child forms containing single and multiple fields.

To ensure that the connector reconciles complex multivalued data, ensure that the child form data is in the same flat file as the parent form data. In other words, every line in the flat file must represent a single record which includes the parent and the child form data. The child form values must be separated by customizable delimiters.

In the following example, the sample multivalued data is presented in the following format:

AccountID,FirstName,LastName,Email,Languages,Roles

```
"111","John","Doe","john.doe@example.com","English;French;Spanish","Administrator
#6-Dec-2013;Backup Operator#7-Nov-2013"
```

Here, Languages and Roles are multivalued data. Languages is a multivalued field without subfields. Roles is a complex multivalued field with subfields like *ROLENAME#STARTDATE*.

2.6 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.

3

Creating an Application for a Disconnected Resource By Using the Flat File Connector

Learn about onboarding applications for a disconnected resource by using the connector.

You can onboard an application for a disconnected resource 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.

3.1 Navigating to the Create Application Screen for a Disconnected Resource

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 .

3.2 Providing Basic Information for a Disconnected Resource

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. In addition, you can mark an attribute from your flat file as multivalued, add child attributes, and set data types if required.

1. On the Basic Information page, ensure that the **Connector Package** option is selected.
2. From the **Select Bundle** drop-down list, select **Flat File Connector 12.2.1.3.0**.
3. Enter the **Application Name**, **Display Name**, and **Description** for the application.

4. If you are using a Connector Server, then in the Basic Configuration section, select the name of your connector server.
5. In the Advanced Settings section, enter values for the parameters as required.

   Table 3-1 describes each parameter in the Advanced Settings section.

Table 3-1 Advanced Settings Parameters for a Disconnected Resource

Parameter	Mandatory?	Description
Connector Name	Yes	This parameter holds the name of the connector class. **Default value:** `org.identityconnectors.flatfile.FlatFileConnector`
Bundle Name	Yes	This parameter holds the name of the connector bundle package. **Default value:** `org.identityconnectors.flatfile`
Bundle Version	Yes	This parameter hods the version of the connector bundle class. **Default value:** `12.3.0`
textQualifier	Yes	Enter the character which determines the start and end of text in a value. The connector ignores any delimiter within the value qualified by the textQualifier parameter. **Default value: "**
fieldDelimiter	Yes	Enter the delimiter for each field in a row. **Default value:** `,`
subFieldDelimiter	Yes, if you are creating a Target application and your flat file contains complex multivalued fields.	Enter the delimiter that separates each subfield within a multivalued field. **Default value:** `#`
multiValueDelimiter	Yes, if you are creating a Target application and your flat file contains complex multivalued fields.	Enter the delimiter that separates each value in a multivalued field. **Default value:** `;`
status Enable Mapping	No	Enter the boolean value which denotes that the account is in the enabled status. Oracle Identity Governance requires the status value to be either True or False. If the boolean value you specify for this parameter is anything other than True or False, then the connector internally maps it to True or False. **Default value:** `Enabled`

Table 3-1 (Cont.) Advanced Settings Parameters for a Disconnected Resource

Parameter	Mandatory?	Description
status Disable Mapping	No	Enter the boolean value which denotes that the account is in the disabled status. Oracle Identity Governance requires the status value to be either True or False. If the boolean value you specify for this parameter is anything other than True or False, then the connector internally maps it to True or False. **Default value:** `Disabled`
System Date Format	No	Enter the format in which date type fields are included in the flat file. **Default value:** `ddmmyy`
flatFileLocation	Yes	Enter the absolute path of the flat file. **Sample value:** `D:\data\ffc\users.csv`

You can also add the following additional attributes depending on your requirement:

- **commentCharacter**: Use this to ignore the processing of lines within the flat file that begin with certain characters such as `#`, `$`, and so on as described in Configuring the Connector to Ignore Comment Characters.
- **headerRowPresent**: Use this to parse CSV files without a header row and set the value of this attribute to `false`.

6. Click **Parse Headers** to parse the headers of your flat file.

   The Flat File Schema Properties table is displayed. This table lists all the attributes present in your flat file and their details such as data type, single-valued or multivalued, mandatory and so on.

7. In the Flat File Schema Properties table, select attributes to designate them as Name and UID attributes as follows:

   a. Select the Name column for an attribute that corresponds to a descriptive name of the account in the flat file that the connector uses for performing reconciliation and update provisioning operations. This value corresponds to the __NAME__ attribute of the connector and is used to generate the reconciliation rule.

   b. Select the UID column for an attribute that corresponds to the unique ID of the account. The connector uses this value to uniquely identify user accounts that it needs to fetch during reconciliation. The connector also uses this value to uniquely identify user accounts during update and delete provisioning operations. This value corresponds to the __UID__ attribute of the connector.

   c. (Optional) Select the Status column for an attribute that denotes the status of the account. The connector uses this attribute during provisioning operations to enable or disable user accounts. In addition, the connector uses this attribute to fetch the status of an account during Status reconciliation. This attribute corresponds to the __ENABLE__ attribute of the connector.

8. If required, change the datatype of an attribute by selecting the required value from the **Data Type** column. For example, for a date attribute, select the **Date** data type. By

default, the data type of all attributes (including attributes that hold date information) displayed in the Flat File Schema Properties table is String.

9. If you are creating a Target application, you can mark any attribute as multivalued by selecting the corresponding checkbox in the MVA column.
10. If you are creating a Target application and your flat file contains complex multivalued attributes, then you must add all its child attributes. For example, if the Department attribute is a complex multivalued attribute, then you must add all its child attributes such as Department Name, Department ID, and so on. To do so:
    1. In the MVA column, select the checkbox corresponding to the complex multivalued attribute.

       The corresponding Add Attribute button in the Complex MVA column is enabled.
    2. Click **Add Attribute** to add its child attributes.
    3. In the text field, enter the child attribute name and then select its datatype from the adjacent drop-down list.
    4. Repeat Steps 8.b and 8.c for adding the rest of the child attributes.
11. Repeat Step 8 for the remaining multivalued attributes in your flat file.
12. Click **Next** to proceed to the Schema page.

3.3 Updating Schema Information for a Disconnected Resource

The Schema page for a Target or an Authoritative application displays the schema of your flat file with mappings with Oracle Identity Governance attributes and flat file attributes. The connector uses these mappings during reconciliation.

If you are creating a Target application, then the table on the Schema page lists the user-specific attribute mappings between the process form fields in Oracle Identity Governance and your flat file attributes. 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 flat file attributes that were designated as Name and UID attributes in the Flat File Schema Properties table on the Basic Information page are marked as the matching key in the Key Field column on the Schema page.

The Schema page also displays the child attribute mappings for any complex multivalued attributes that you may have added to the Flat File Schema Properties table on the Basic Information page. You must ensure that you designate an attribute as the key field to be used for entity matching during reconciliation.

If you are creating an Authoritative application, then the table on the Schema page lists all the flat file attributes and their data types, and specifies whether a given attribute is a mandatory attribute for reconciliation. Note that this page does not list the user attribute mapping between the flat file attributes and reconciliation fields in Oracle Identity Governance. Therefore, you must map each flat file attribute listed in the Target Attribute column with the corresponding Oracle Identity Governance field in the Identity Display Name column.

Perform the following procedure on the Schema page:

1. Review the user-specific attribute mappings and then in the Mandatory column, select or deselect the checkbox corresponding to an attribute to specify whether it is mandatory for reconciliation, if required.
2. For any target system attribute of the Date data type that must be mapped to a String type attribute in Oracle Identity Governance, click the ☰ icon and select the **Date** checkbox.
3. If you are creating a Target application, then in the child attribute mapping table:
   a. Select the **Key Field** column for an attribute that the connector must use for entity matching during reconciliation.
   b. If required, click the ☰ icon to add additional properties to the child attribute such as Lookup, Date, or Entitlements.

   > **Note:**
   >
   > If you have provided additional properties such as Lookup and Entitlements, then ensure that you note the name of the lookup. You must provide this lookup name in the reconciliation job for Entitlements.

4. If you are creating an Authoritative application, then in the Identity Display Name column, enter the name of the OIG User form field to which a given flat file attribute in the Target Attribute column must be mapped to. Repeat this for all the attributes in the table.
5. Click **Next** to proceed to the Settings page.

3.4 Providing Settings Information for a Disconnected Resource

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. Ensure that you enter values for the mandatory parameters (marked by the asterisk (*) symbol) of all the reconciliation jobs and then click **Next** to proceed to the Finish page.

The following are the reconciliation jobs that will be available for use after the application is created:

- Flat File Full
- Flat File Diff Sync
- Flat File Delete Sync
- Flat File Entitlement

> **Note:**
>
> The Flat File Entitlement reconciliation job is available if you are creating a Target application.

- Flat File Delete

Flat File Full

The Flat File Accounts Loader reconciliation job is used for reconciling accounts from a flat file and creating corresponding accounts in Oracle Identity Governance.

Table 3-2 describes the parameters of the Flat File Accounts Loader reconciliation job.

Table 3-2 Parameters of the Flat File Accounts Loader

Parameter	Description
FlatFile Instance Name	This parameter holds the name of the application for your flat file. This value is the same as the value that you provided for the **Application Name** field while creating the flat file application.
Flat File directory	Enter the name and complete path to the directory containing the flat file that the connector needs to parse. **Note:** The OIG administrator must have read and write permissions on this directory.
Archive directory	Enter the name of the directory in which the processed flat files must be saved. If you do not specify a value for this attribute, the connector creates a directory named "archived" within the directory containing the flat file, and the processed files are saved in this location. **Note:** The OIG administrator must have read and write permissions on this directory to enable adding of the processed flat files to the archive directory.
Filter	Enter the expression for filtering records that the scheduled job must reconcile. Sample value: `startsWith('email','john')` For information about the filters expressions that you can create and use, see ICF Filter Syntax in *Developing and Customizing Applications for Oracle Identity Governance*.
Incremental Recon Attribute	Enter the name of the flat file column that holds the time stamp at which the record was last modified. The value in this attribute is used during incremental reconciliation to determine the newest or latest record reconciled from the flat file. **Sample value:** `LastUpdated`

Table 3-2 (Cont.) Parameters of the Flat File Accounts Loader

Parameter	Description
Latest Token	This parameter holds the value of the Incremental Recon Attribute. **Note:** The reconciliation engine automatically enters a value for this attribute after execution. It is recommended that you do not change the value of this attribute. If you manually specify a value for this attribute, then the connector reconciles only user accounts that have been modified after the time stamp specified as the value of this parameter. If you want to perform a full reconciliation run, then clear the value in this field.
Scheduled Task Name	This parameter holds the name of the scheduled job. **Default value:** *APP_NAME* Flat File Accounts Loader Here, *APP_NAME* is the Application Name you provided while creating the application.

Flat File Diff Sync

The Flat File Accounts Diff Sync Reconciliation job is used for performing diff-based reconciliation.

This reconciliation job compares the two flat files and returns the deleted accounts alone. This reconciliation job is used to detect deleted accounts from flat files for enterprise target systems that do not support the export of only the deleted accounts. The following are the two flat file directories that are the input for these scheduled jobs:

- Previous Flat File directory

  This is the flat file containing all the accounts before delete.

- Current Flat File directory

  This is the flat file that is exported from the enterprise target system after accounts have been deleted in the enterprise target system.

When you run this reconciliation job, the connector will detect the accounts that are missing in the current flat file by comparing them with the accounts in the previous flat file, and will generate delete reconciliation events only for the missing accounts.

Table 3-3 describes the parameters of the Flat File Accounts Diff Sync Reconciliation job.

Table 3-3 Parameters of the Flat File Accounts Diff Sync Reconciliation Job

Parameter	Description
FlatFile Instance Name	This parameter holds the name of the application for your flat file. This value is the same as the value that you provided for the **Application Name** field while creating the flat file application.

Table 3-3 (Cont.) Parameters of the Flat File Accounts Diff Sync Reconciliation Job

Parameter	Description
Previous Flat File directory	Enter the name and complete path of the flat file directory that contains the records from the enterprise target system that were present previously.
Current Flat File directory	Enter the name and complete path of the flat file directory that contains the current records from the enterprise target system.
Archive directory	Enter the name of the directory in which the processed flat files must be saved. If you do not specify a value for this attribute, the connector creates a directory named "archived" within the directory containing the flat file, and the processed files are saved in this location. **Note:** The OIG administrator must have read and write permissions on this directory to enable adding of the processed flat files to the archive directory.
Sync Token	If you are using this reconciliation job for the first time, do not specify a value for this parameter. For subsequent runs, the reconciliation engine automatically enters a value for this parameter. Sample value: `<String>123454502019<String>`
Scheduled Task Name	This parameter holds the name of the scheduled job. **Default value:** *APP_NAME* Flat File Accounts Diff Sync Reconciliation Here, *APP_NAME* is the Application Name you provided while creating the application.

Flat File Delete Sync

The Flat File Accounts Delete Sync Reconciliation job is used to perform a delete reconciliation run.

If you want to perform a filtered delete reconciliation run based on any field in the flat file, then specify a value for the following attributes of the scheduled job:

- Delete Attribute
- Delete Attribute Value

If you do not specify a value for the preceding attributes, then all the records in the flat file are considered as deleted records.

Table 3-4 describes the parameters of the Flat File Accounts Delete Sync Reconciliation job.

Table 3-4   Parameters of the Flat File Accounts Delete Sync Reconciliation Job

Parameter	Description
FlatFile Instance Name	This parameter holds the name of the application for your flat file. This value is the same as the value that you provided for the **Application Name** field while creating the flat file application.
Flat File directory	Enter the name and complete path to the directory containing the flat file that the connector needs to parse. **Note:** The OIG administrator must have read and write permissions on this directory.
Archive directory	Enter the name of the directory in which the processed flat files must be saved. If you do not specify a value for this attribute, the connector creates a directory named "archived" within the directory containing the flat file, and the processed files are saved in this location. **Note:** The OIG administrator must have read and write permissions on this directory to enable adding of the processed flat files to the archive directory.
Delete Attribute	Enter the name of the column in the flat file that represents whether an account is deleted or not. Enter a value for this attribute if you want to perform filtered delete reconciliation. Default value: None Sample value: `isDeleted`
Delete Attribute Value	Enter the value that is mentioned in the column that specifies whether an account is deleted. This column is the value that you specified as the value of the Delete Attribute parameter. Sample value: `Yes`
Sync Token	If you are using this schedule job for the first time, do not specify a value for this attribute. For subsequent runs, the reconciliation engine automatically enters a value for this attribute. Sample value: `<String>123454502019<String>`
Scheduled Task Name	This parameter holds the name of the scheduled job. **Default value:** *APP_NAME* Flat File Accounts Delete Sync Reconciliation Here, *APP_NAME* is the Application Name you provided while creating the application.

Flat File Entitlement

The Flat File Entitlements Loader reconciliation job is used to reconcile both lookup values and entitlements from a flat file.

In addition to reconciling the lookups from a flat file, this reconciliation job also adds the entitlements for lookups that are associated with an Entitlement, and synchronizes the catalog with the entitlements automatically. The Flat File Entitlements Loader reconciliation job also supports full and incremental reconciliation of lookup values and entitlements.

Table 3-5 describes the parameters of the Flat File Entitlements Loader reconciliation job.

Table 3-5 Parameters of the Flat File Entitlements Loader Reconciliation Job

Parameter	Description
FlatFile Instance Name	This parameter holds the name of the application for your flat file. This value is the same as the value that you provided for the **Application Name** field while creating the flat file application.
Flat File directory	Enter the name and complete path to the directory containing the flat file for your entitlements that the connector needs to parse. **Note:** The OIG administrator must have read and write permissions on this directory.
Archive directory	Enter the name of the directory in which the processed flat files must be saved. If you do not specify a value for this attribute, the connector creates a directory named "archived" within the directory containing the flat file, and the processed files are saved in this location. **Note:** The OIG administrator must have read and write permissions on this directory to enable adding of the processed flat files to the archive directory.
Lookup Name	Enter the name of the lookup definition into which the connector must load all the values fetched from the flat file for entitlements. **Note:** The name of this lookup definition must be the same as the one that you specified on the Schema page for Entitlement lookup. **Sample value:** `Lookup.FlatFile.Entitlements`
Code Key Attribute	Enter the name of the flat file attribute whose values you want to populate into the Code Key column the of lookup definition specified as the value of the Lookup Name parameter. **Default value:** `__NAME__`
Decode Attribute	Enter the name of the flat file attribute whose values you want to populate into the Decode column of the lookup definition specified as the value of the Lookup Name parameter. **Default value:** `__NAME__`

Table 3-5 (Cont.) Parameters of the Flat File Entitlements Loader Reconciliation Job

Parameter	Description
Mode	Enter the mode in which the job must run. The possible value for this parameter are: • `Entitlement` - Use this value to perform entitlement lookup reconciliation. For example, to perform group or roles lookup field synchronization. • `Full` - Use this value to reconcile all user accounts from your flat file. In other words, use this to perform a full reconciliation. • `Delete` - Use this value to delete all revoked user accounts. **Default value:** `Full`
Is Entitlement?	Enter `true` if the lookup definition is linked to an Entitlement field (for example, Roles). Enter `false` if the lookup name in the flat file is a plain lookup field (for example, Languages). This flag will decide if the ENT_LIST and Catalog should be updated with the lookup values. **Default value:** `true`

Flat File Delete

The Flat File Accounts Delete Reconciliation job is used to reconcile data about deleted accounts. During a reconciliation run, for each account deleted on the enterprise target system, the corresponding OIG account is deleted.

Use this reconciliation job if you cannot export flat files containing only a list of deleted accounts, but can periodically export flat files containing all accounts in the enterprise target system.

Note:

This process is resource consuming as Oracle Identity Governance has to verify all the records from the flat file and compare it with existing records to identify whether each record has been deleted or not.

Table 3-6 describes the parameters of the Flat File Accounts Delete Reconciliation job.

Table 3-6 Parameters of the Flat File Accounts Delete Reconciliation Job

Parameter	Description
FlatFile Instance Name	This parameter holds the name of the application for your flat file. This value is the same as the value that you provided for the **Application Name** field while creating the flat file application.
Flat File directory	Enter the name and complete path to the directory containing the flat file that the connector needs to parse. **Note:** The OIG administrator must have read and write permissions on this directory.
Archive directory	Enter the name of the directory in which the processed flat files must be saved. If you do not specify a value for this attribute, the connector creates a directory named "archived" within the directory containing the flat file, and the processed files are saved in this location. **Note:** The OIG administrator must have read and write permissions on this directory to enable adding of the processed flat files to the archive directory.

3.5 Reviewing and Submitting the Application Details for a Disconnected Resource

On the Finish page, review your application summary and click **Finish** to submit the application.

On the Finish page, review the summary of the Target or Authoritative application you are creating. If required, click **Back** to make any changes to the application details. If no changes are required, click **Finish** to submit the application details. The application is created in Oracle Identity Governance.

When you are prompted whether you want to create a default request form, click **Yes** or **No**. If you click **Yes**, then the default form is automatically created and is attached with the newly created application. The default form is created with the same name as the application. You cannot modify the default form later. Therefore, if you want to customize it, click **No** to manually create a new form and attach it with your application.

4

Configuring an Application for a Connected Resource By Using the Flat File Connector

Learn about configuring and managing flat files for a connected resource by using the connector.

- Configuring an Application for a Connected Resource
- Managing Flat File Configurations for a Connected Application

4.1 Configuring an Application for a Connected Resource

You can configure a flat file application for a connected resource and load entities into Oracle Identity Governance from the connector package by navigating to the Configure Application screen.

- Navigating to the Configure Application Screen
- Providing Basic Information for a Connected Resource
- Providing Settings Information for a Connected Resource

4.1.1 Navigating to the Configure Application Screen

To navigate to the Configure Application screen, you must log in to Identity Self Service and then choose the Applications box on the Manage tab. Then, on the Applications page, search for and configure the application into which you want to load entities.

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, search for and select the application into which you want to load entities as follows: For example, search for and select an application named SFApp for the Salesforce connector.
   a. In the **Search** field, select the **Name** attribute from the drop-down list and then enter the search criterion.
   b. From the results that are returned in the table, select the application.
3. Click **Flat File** on the toolbar, and then select **Configure**.

The Configure Application *APP_NAME* using Flat File screen with the Basic Information page is displayed. Here, *APP_NAME* is the name of the application into which you want to load entities.

4.1.2 Providing Basic Information for a Connected Resource

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 configuration name, display name and its description, and info that the connector uses during reconciliation. In addition, you can mark an attribute from your flat file as multivalued, add child attributes, and set data types if required.

1. On the Basic Information page, enter values for the following:
   - **Configuration Name**: Enter a unique name for an application that you want to configure. This is a mandatory field.
   - **Display Name**: Enter the display name for the application that you are creating. This is a mandatory field.
   - **Description**: Enter text that describes the application being created. This is an optional field.
   - **Parent Application Name**: Select the name of the application on which the current application for which you are configuring the flat file has a dependency on. For example, if you are configuring a flat file application for a Microsoft Exchange application, then you must select the Microsoft Active Directory application. This is because the Microsoft Exchange application has a dependency on Microsoft Active Directory application.
2. If you are using a Connector Server, then in the Basic Configuration section, select the name of your connector server.
3. In the Advanced Settings section, enter values for the parameters as required.

   Table 4-1 describes each parameter in the Advanced Settings section.

Table 4-1 Advanced Settings Parameters for a Connected Resource

Parameter	Mandatory?	Description
Connector Name	Yes	This parameter holds the name of the connector class. **Default value:** `org.identityconnectors.flatfile.FlatFileConnector`
Bundle Name	Yes	This parameter holds the name of the connector bundle package. **Default value:** `org.identityconnectors.flatfile`
Bundle Version	Yes	This parameter hods the version of the connector bundle class. **Default value:** `12.3.0`

Table 4-1 (Cont.) Advanced Settings Parameters for a Connected Resource

Parameter	Mandatory?	Description
textQualifier	Yes	Enter the character which determines the start and end of text in a value. The connector ignores any delimiter within the value qualified by the textQualifier parameter. **Default value:** "
fieldDelimiter	Yes	Enter the delimiter for each field in a row. **Default value:** `,`
subFieldDelimiter	Yes, if your flat file contains complex multivalued fields.	Enter the delimiter that separates each subfield within a multivalued field. **Default value:** `#`
multiValueDelimiter	Yes, if your flat file contains complex multivalued fields.	Enter the delimiter that separates each value in a multivalued field. **Default value:** `;`
status Enable Mapping	No	Enter the boolean value which denotes that the account is in the enabled status. Oracle Identity Governance requires the status value to be either True or False. If the boolean value you specify for this parameter is anything other than True or False, then the connector internally maps it to True or False. **Default value:** `Enabled`
status Disable Mapping	No	Enter the boolean value which denotes that the account is in the disabled status. Oracle Identity Governance requires the status value to be either True or False. If the boolean value you specify for this parameter is anything other than True or False, then the connector internally maps it to True or False. **Default value:** `Disabled`
System Date Format	No	Enter the format in which date type fields are included in the flat file. **Default value:** `ddmmyy`
flatFileLocation	Yes	Enter the absolute path of the flat file. **Sample value:** `D:\data\ffc\users.csv`

You can also add the following additional attributes depending on your requirement:

- **commentCharacter**: Use this to ignore the processing of lines within the flat file that begin with certain characters such as `#`, `$`, and so on as described in Configuring the Connector to Ignore Comment Characters.
- **headerRowPresent**: Use this to parse CSV files without a header row and set the value of this attribute to `false`.

4. Click **Parse Headers** to parse the headers of your flat file.

   The Flat File Schema Properties table is displayed. This table lists all the attributes present in your flat file and their details such as data type, single-valued or multivalued, mandatory and so on.

5. In the Flat File Schema Properties table, select attributes to designate them as Name and UID attributes as follows:

   a. Select the Name column for an attribute that corresponds to a descriptive name of the account in the flat file that the connector uses for performing reconciliation and update provisioning operations. This value corresponds to the __NAME__ attribute of the connector and is used to generate the reconciliation rule.

   b. Select the UID column for an attribute that corresponds to the unique ID of the account. The connector uses this value to uniquely identify user accounts that it needs to fetch during reconciliation. The connector also uses this value to uniquely identify user accounts during update and delete provisioning operations. This value corresponds to the __UID__ attribute of the connector.

   c. (Optional) Select the Status column for an attribute that denotes the status of the account. The connector uses this attribute during provisioning operations to enable or disable user accounts. In addition, the connector uses this attribute to fetch the status of an account during Status reconciliation. This attribute corresponds to the __ENABLE__ attribute of the connector.

6. If required, change the datatype of an attribute by selecting the required value from the **Data Type** column. By default, the data type of all attributes (including attributes that hold date information) displayed in the Flat File Schema Properties table is String.

7. If you are configuring the flat file for a Target application, then you can mark any attribute as multivalued by selecting the corresponding checkbox in the MVA column, if required.

8. If you are configuring the flat file for a Target application and your flat file contains complex multivalued attributes, then you must add all its child attributes. For example, if the Department attribute is a complex multivalued attribute, then you must add all its child attributes such as Department Name, Department ID, and so on. To do so:

   a. In the MVA column, select the checkbox corresponding to the complex multivalued attribute.

      The corresponding Add Attribute button in the Complex MVA column is enabled.

   b. Click **Add Attribute** to add its child attributes.

   c. In the text field, enter the child attribute name and then select its datatype from the adjacent drop-down list.

   d. Repeat Steps 8.b and 8.c for adding the rest of the child attributes.

9. Repeat Step 8 for the remaining multivalued attributes in your flat file.

10. Click **Apply** to apply all the information provided so far.

4.1.3 Providing Settings Information for a Connected Resource

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 jobs that you do not want to use. If required, you can also add new reconciliation jobs to meet your requirements.

Ensure that you enter values for the mandatory parameters (marked by the asterisk (*) symbol) of all the reconciliation jobs and then click **Apply**. A message stating that the flat file configuration was created successfully is displayed.

The following are the reconciliation jobs that will be available for use after the application is created:

- Flat File Full
- Flat File Diff Sync
- Flat File Delete Sync
- Flat File Entitlement
- Flat File Delete

Flat File Full

The Flat File Accounts Loader reconciliation job is used for reconciling accounts from a flat file and creating corresponding accounts in Oracle Identity Governance. Use this reconciliation job for performing a full or incremental reconciliation run.

Table 4-2 describes the parameters of the Flat File Accounts Loader reconciliation job.

Table 4-2 Parameters of the Flat File Accounts Loader Job

Parameter	Description
FlatFile Instance Name	This parameter holds the connector display name that is displayed on the Basic Information page. **Default value:** `Flat File Connector`
Flat File directory	Enter the name and complete path to the directory containing the flat file that the connector needs to parse. **Note:** The OIG administrator must have read and write permissions on this directory.

Table 4-2   (Cont.) Parameters of the Flat File Accounts Loader Job

Parameter	Description
Archive directory	Enter the name of the directory in which the processed flat files must be saved. If you do not specify a value for this attribute, the connector creates a directory named "archived" within the directory containing the flat file, and the processed files are saved in this location. **Note:** The OIG administrator must have read and write permissions on this directory to enable adding of the processed flat files to the archive directory.
Filter	Enter the expression for filtering records that the reconciliation job must retrieve. Sample value: `startsWith('email','john')` For information about the filters expressions that you can create and use, see ICF Filter Syntax in *Developing and Customizing Applications for Oracle Identity Governance*.
Incremental Recon Attribute	Enter the name of the flat file column that holds the time stamp at which the record was last modified. The value in this attribute is used during incremental reconciliation to determine the newest or latest record reconciled from the flat file. **Sample value:** `LastUpdated`
Latest Token	This parameter holds the value of the Incremental Recon Attribute. **Note:** The reconciliation engine automatically enters a value for this attribute after execution. It is recommended that you do not change the value of this attribute. If you manually specify a value for this attribute, then the connector reconciles only user accounts that have been modified after the time stamp specified as the value of this parameter. If you want to perform a full reconciliation run, then clear the value in this field.
Scheduled Task Name	This parameter holds the name of the scheduled job. **Default value:** `Flat File Accounts Loader`

Flat File Diff Sync

The Flat File Accounts Diff Sync Reconciliation job is used for performing diff-based reconciliation.

This reconciliation job compares the two flat files and returns the deleted accounts alone. This reconciliation job is used to detect deleted accounts from flat files for enterprise target systems that do not support the export of only the deleted accounts. The following are the two flat file directories that are the input for these scheduled jobs:

- Previous Flat File directory

  This is the flat file containing all the accounts before delete.

- Current Flat File directory

  This is the flat file that is exported from the enterprise target system after accounts have been deleted in the enterprise target system.

When you run this reconciliation job, the connector will detect the accounts that are missing in the current flat file by comparing them with the accounts in the previous flat file, and will generate delete reconciliation events only for the missing accounts.

Table 4-3 describes the parameters of the Flat File Accounts Diff Sync Reconciliation job.

Table 4-3 Parameters of the Flat File Accounts Diff Sync Reconciliation Job

Parameter	Description
FlatFile Instance Name	This parameter holds the connector display name that is displayed on the Basic Information page. **Default value:** `Flat File Connector`
Previous Flat File directory	Enter the name and complete path of the flat file directory that contains the records from the enterprise target system that were present previously.
Current Flat File directory	Enter the name and complete path of the flat file directory that contains the current records from the enterprise target system.
Archive directory	Enter the name of the directory in which the processed flat files must be saved. If you do not specify a value for this attribute, the connector creates a directory named "archived" within the directory containing the flat file, and the processed files are saved in this location. **Note:** The OIG administrator must have read and write permissions on this directory to enable adding of the processed flat files to the archive directory.
Sync Token	If you are using this reconciliation job for the first time, do not specify a value for this parameter. For subsequent runs, the reconciliation engine automatically enters a value for this parameter. Sample value: `<String>123454502019<String>`
Scheduled Task Name	This parameter holds the name of the scheduled job. **Default value:** `Flat File Accounts Diff Sync Reconciliation`

Flat File Delete Sync

The Flat File Accounts Delete Sync Reconciliation job is used to perform a delete reconciliation run.

If you want to perform a filtered delete reconciliation run based on any field in the flat file, then specify a value for the following attributes of the scheduled job:

- Delete Attribute
- Delete Attribute Value

If you do not specify a value for the preceding attributes, then all the records in the flat file are considered as deleted records.

Table 4-4 describes the parameters of the Flat File Accounts Delete Sync Reconciliation job.

Table 4-4 Parameters of the Flat File Accounts Delete Sync Reconciliation Job

Parameter	Description
FlatFile Instance Name	This parameter holds the connector display name that is displayed on the Basic Information page. **Default value:** `Flat File Connector`
Flat File directory	Enter the name and complete path to the directory containing the flat file that the connector needs to parse. **Note:** The OIG administrator must have read and write permissions on this directory.
Archive directory	Enter the name of the directory in which the processed flat files must be saved. If you do not specify a value for this attribute, the connector creates a directory named "archived" within the directory containing the flat file, and the processed files are saved in this location. **Note:** The OIG administrator must have read and write permissions on this directory to enable adding of the processed flat files to the archive directory.
Delete Attribute	Enter the name of the column in the flat file that represents whether an account is deleted or not. Enter a value for this attribute if you want to perform filtered delete reconciliation. Default value: None Sample value: `isDeleted`
Delete Attribute Value	Enter the value that is mentioned in the column that specifies whether an account is deleted. This column is the value that you specified as the value of the Delete Attribute parameter. Sample value: `Yes`
Sync Token	If you are using this schedule job for the first time, do not specify a value for this attribute. For subsequent runs, the reconciliation engine automatically enters a value for this attribute. Sample value: `<String>123454502019<String>`
Scheduled Task Name	This parameter holds the name of the scheduled job. **Default value:** `Flat File Accounts Delete Sync Reconciliation`

Flat File Entitlement

The Flat File Entitlements Loader reconciliation job is used to reconcile both lookup values and entitlements from a flat file.

In addition to reconciling the lookups from a flat file, this reconciliation job also adds the entitlements for lookups that are associated with an Entitlement, and synchronizes the catalog with the entitlements automatically. The Flat File Entitlements Loader reconciliation job also supports full and incremental reconciliation of lookup values and entitlements.

Table 4-5 describes the parameters of the Flat File Entitlements Loader reconciliation job.

Table 4-5 Parameters of the Flat File Entitlements Loader Reconciliation Job

Parameter	Description
FlatFile Instance Name	This parameter holds the connector display name that is displayed on the Basic Information page. **Default value:** `Flat File Connector`
Flat File directory	Enter the name and complete path to the directory containing the flat file for your entitlements that the connector needs to parse. **Note:** The OIG administrator must have read and write permissions on this directory.
Archive directory	Enter the name of the directory in which the processed flat files must be saved. If you do not specify a value for this attribute, the connector creates a directory named "archived" within the directory containing the flat file, and the processed files are saved in this location. **Note:** The OIG administrator must have read and write permissions on this directory to enable adding of the processed flat files to the archive directory.
Lookup Name	Enter the name of the lookup definition into which the connector must load all the values fetched from the flat file for entitlements. **Note:** The name of this lookup definition must be the same as the one that you specified on the Schema page for Entitlement lookup. **Sample value:** `Lookup.FlatFile.Entitlements`
Code Key Attribute	Enter the name of the flat file attribute whose values you want to populate into the Code Key column the of lookup definition specified as the value of the Lookup Name parameter.
Decode Attribute	Enter the name of the flat file attribute whose values you want to populate into the Decode column of the lookup definition specified as the value of the Lookup Name parameter.

Table 4-5 (Cont.) Parameters of the Flat File Entitlements Loader Reconciliation Job

Parameter	Description
Mode	Enter the mode in which the job must run. The possible value for this parameter are: • `Entitlement` - Use this value to perform entitlement lookup reconciliation. For example, to perform group or roles lookup field synchronization. • `Full` - Use this value to reconcile all user accounts from your flat file. In other words, use this to perform a full reconciliation. • `Delete` - Use this value to delete all revoked user accounts. **Default value:** `Full`
Is Entitlement?	Enter `true` if the lookup definition is linked to an Entitlement field (for example, Roles). Enter `false` if the lookup name in the flat file is a plain lookup field (for example, Languages). This flag will decide if the ENT_LIST and Catalog should be updated with the lookup values. **Default value:** `true`

Flat File Delete

The Flat File Accounts Delete Reconciliation job is used to reconcile data about deleted accounts. During a reconciliation run, for each account deleted on the enterprise target system, the corresponding OIG account is deleted.

Use this reconciliation job if you cannot export flat files containing only a list of deleted accounts, but can periodically export flat files containing all accounts in the enterprise target system.

> **Note:**
>
> This process is resource consuming as Oracle Identity Governance has to verify all the records from the flat file and compare it with existing records to identify whether each record has been deleted or not.

Table 4-6 describes the parameters of the Flat File Accounts Delete Reconciliation job.

Table 4-6 Parameters of the Flat File Accounts Delete Reconciliation Job

Parameter	Description
FlatFile Instance Name	This parameter holds the connector display name that is displayed on the Basic Information page. **Default value:** `Flat File Connector`

Table 4-6 (Cont.) Parameters of the Flat File Accounts Delete Reconciliation Job

Parameter	Description
Flat File directory	Enter the name and complete path to the directory containing the flat file that the connector needs to parse. **Note:** The OIG administrator must have read and write permissions on this directory.
Archive directory	Enter the name of the directory in which the processed flat files must be saved. If you do not specify a value for this attribute, the connector creates a directory named "archived" within the directory containing the flat file, and the processed files are saved in this location. **Note:** The OIG administrator must have read and write permissions on this directory to enable adding of the processed flat files to the archive directory.

4.2 Managing Flat File Configurations for a Connected Application

You can edit any flat file configuration that you have created for a connected application by using the **Manage** option for Flat Files.

See Managing Flat File Configurations in *Performing Self Service Tasks with Oracle Identity Governance* for more information.

5

Performing the Postconfiguration Tasks for the Flat File Connector

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

- Configuring Oracle Identity Governance
- Managing Logging
- Localizing Field Labels in UI Forms
- Configuring the Connector to Ignore Comment Characters
- Creating the IT Resource for the Connector Server

5.1 Configuring Oracle Identity Governance

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 configure Oracle Identity Governance:

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

5.1.1 Creating and Activating a Sandbox

You must create and activate a sandbox to begin using the customization and form management features. You can then publish the sandbox to make the customizations available to other users.

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

5.1.2 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.

5.1.3 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*.

5.1.4 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*

5.2 Managing Logging

Oracle Identity Governance uses Oracle Java Diagnostic Logging (OJDL) for logging. OJDL is based on java.util.logger.

The following topics provide detailed information about logging:

- Understanding Log Levels
- Enabling Logging

5.2.1 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 5-1.

Table 5-1 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

Table 5-1 (Cont.) Log Levels and ODL Message Type:Level Combinations

Java Level	ODL Message Type:Level
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.

5.2.2 Enabling Logging

Perform this procedure to enable logging in Oracle WebLogic Server.

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

```
<log_handler name='flatfile-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.flatfile" level="[LOG_LEVEL]"
useParentHandlers="false">
     <handler name="flatfile-handler"/>
     <handler name="console-handler"/>
   </logger>
```

   2. Replace both occurrences of **[LOG_LEVEL]** with the ODL message type and level combination that you require. Table 5-1 lists the supported message type and level combinations.

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

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

```
<log_handler name='flatfile-handler' level='NOTIFICATION:1'
class='oracle.core.ojdl.logging.ODLHandlerFactory'>
<property name='logreader:' value='off'/>
     <property name='path'
value='F:\MyMachine\middleware\user_projects\domains\base_domain1\servers
\oim_server1\logs\oim_server1-diagnostic-1.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.flatfile" level="NOTIFICATION:1"
   useParentHandlers="false">
        <handler name="flatfile-handler"/>
        <handler name="console-handler"/>
      </logger>
```

With these sample values, when you use Oracle Identity Governance, 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.

5.3 Localizing Field Labels in UI Forms

Perform this procedure to localize field labels that are added to UI forms.

1. Log in to Oracle Enterprise Manager.
2. In the left pane, expand **Application Deployments** and then select **oracle.iam.console.identity.sysadmin.ear.**
3. In the right pane, from the Application Deployment list, select **MDS Configuration.**
4. On the MDS Configuration page, click **Export** and save the archive (oracle.iam.console.identity.sysadmin.ear_V2.0_metadata.zip) to the local computer.
5. Extract the contents of the archive, and open the following file in a text editor:

   SAVED_LOCATION\xliffBundles\oracle\iam\ui\runtime\BizEditorBundle_en.xlf

   > **Note:**
   >
   > You will not be able to view the BizEditorBundle.xlf file unless you complete creating the application for your target system or perform any customization such as creating a UDF.

6. 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 ACME application instance. The original code is:

```
<trans-unit id="$
{adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceBundl
e']
['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.userEO.
UD_ACME_LANGUAGE__c_description']}">
<source>Language</source>
</target>
</trans-unit>
<trans-unit
id="sessiondef.oracle.iam.ui.runtime.form.model.ACME.entity.ACMEEO.UD_ACM
E_LANGUAGE__c_LABEL">
<source>Language</source>
</target>
</trans-unit>
```

d. Update the <target> element of trans-unit shown in Step 6.b with the localized string of the field name as follows:

```
<trans-unit id="$
{adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceBundl
e']
['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.userEO.
UD_ACME_LANGUAGE__c_description']}">
<source>Language</source>
<target>\u8A00\u8A9E</target>
</trans-unit>
<trans-unit
id="sessiondef.oracle.iam.ui.runtime.form.model.ACME.entity.ACMEEO.UD_ACM
E_LANGUAGE__c_LABEL">
<source>Language</source>
<target>\u8A00\u8A9E</target>
</trans-unit>
```

e. Repeat Steps 6.a through 6.d for all attributes of the process form.

f. 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.

7. 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

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

5.4 Configuring the Connector to Ignore Comment Characters

You can configure the connector to ignore the processing of lines that begin with certain characters like #, $, and so on.

These configurable characters are considered as comment characters, and sentences beginning with such characters are considered as comments. The connector implementation will skip the lines that start with the configured comment character.

You can configure this by adding an attribute named commentCharacter in the Advanced Settings section of your flat file application. To do so, click **Add Attribute** and then in the New Attribute window that is displayed, enter values for the following fields and click **OK**:

- **Name**: Enter `commentCharacter` as the name of the attribute being added.
- **Value**: Enter the characters that the connector must ignore. For example, `#`, `$`, and so on.
- **Category**: You need not enter any value for this field. The default value for this field is Custom.
- **Display Name**: Enter a display name for the attribute being added. For example, `Comment Character`.

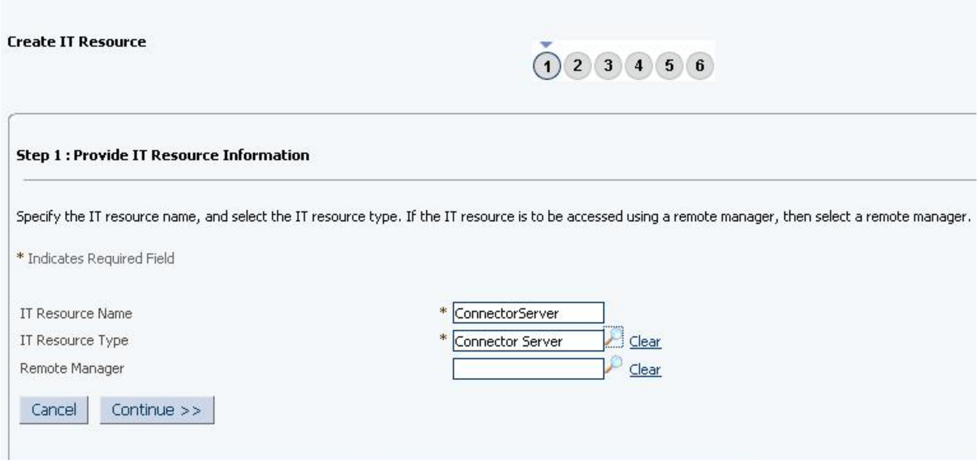
5.5 Creating the IT Resource for the Connector Server

You must create an IT resource for the Connector Server if you have deployed the connector bundle remotely in a Connector Server.

To create the IT resource for the Connector Server:

1. Log in to Oracle Identity System Administration.
2. In the left pane, under Configuration, click **IT Resource.**
3. In the Manage IT Resource page, click **Create IT Resource.**
4. On the Step 1: Provide IT Resource Information page, perform the following steps:
   - **IT Resource Name**: Enter a name for the IT resource.
   - **IT Resource Type**: Select **Connector Server** from the IT Resource Type list.
   - **Remote Manager**: Do not enter a value in this field.
5. Click **Continue**. Figure 5-1 shows the IT resource values added on the Create IT Resource page.

Figure 5-1 Step 1: Provide IT Resource Information



Create IT Resource

Step 1 : Provide IT Resource Information

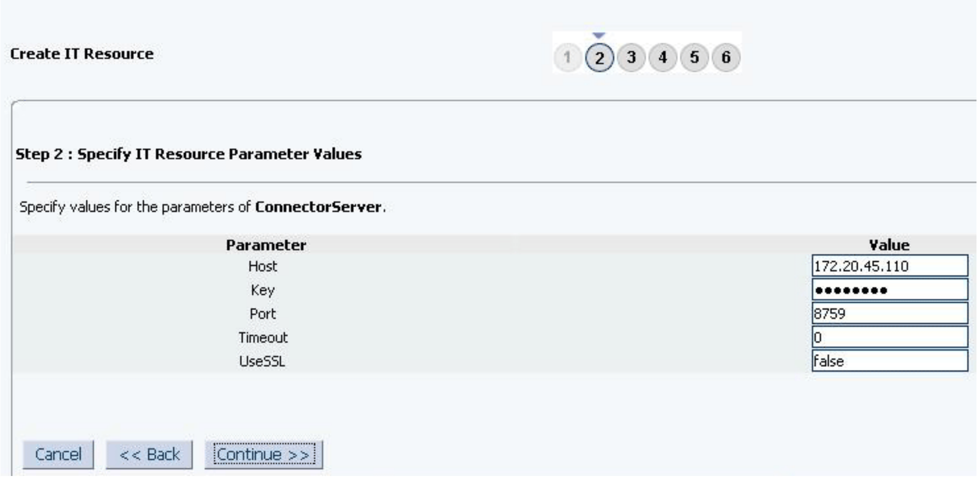
Specify the IT resource name, and select the IT resource type. If the IT resource is to be accessed using a remote manager, then select a remote manager.

* Indicates Required Field

IT Resource Name	* ConnectorServer
IT Resource Type	* Connector Server
Remote Manager	

6. On the Step 2: Specify IT Resource Parameter Values page, specify values for the parameters of the IT resource and then click **Continue**. Figure 5-2 shows the Step 2: Specify IT Resource Parameter Values page.

Figure 5-2 Step 2: Specify IT Resource Parameter Values



Create IT Resource

Step 2 : Specify IT Resource Parameter Values

Specify values for the parameters of **ConnectorServer**.

Parameter	Value
Host	172.20.45.110
Key	••••••••
Port	8759
Timeout	0
UseSSL	false

Figure 5-5 provides information about the parameters of the IT resource.

Table 5-2 Parameters of the IT Resource for the Connector Server

Parameter	Description
Host	Enter the host name or IP address of the computer hosting the connector server. Sample value: `RManager`
Key	Enter the key for the Java connector server.
Port	Enter the number of the port at which the connector server is listening. Default value: `8759`

Table 5-2 (Cont.) Parameters of the IT Resource for the Connector Server

Parameter	Description
Timeout	Enter an integer value which specifies the number of milliseconds after which the connection between the connector server and Oracle Identity Governance times out. Sample value: `300`
UseSSL	Enter `true` to specify that you will configure SSL between Oracle Identity Governance and the Connector Server. Otherwise, enter `false`. Default value: `false` **Note:** It is recommended that you configure SSL to secure communication with the connector server. To configure SSL, run the connector server by using the /setKey [*key*] option. The value of this key must be specified as the value of the Key IT resource parameter of the connector server.

7. On the Step 3: Set Access Permission to IT Resource page, the `SYSTEM ADMINISTRATORS` group is displayed by default in the list of groups that have Read, Write, and Delete permissions on the IT resource that you are creating.

   > **Note:**
   >
   > This step is optional.

   If you want to assign groups to the IT resource and set access permissions for the groups, then:

   a. Click **Assign Group**.

   b. For the groups that you want to assign to the IT resource, select **Assign** and the access permissions that you want to set. For example, if you want to assign the `ALL USERS` group and set the Read and Write permissions to this group, then you must select the respective check boxes in the row, as well as the Assign check box, for this group.

   c. Click **Assign**.

8. On the Step 3: Set Access Permission to IT Resource page, if you want to modify the access permissions of groups assigned to the IT resource, then:

   > **Note:**
   >
   > - This step is optional.
   > - You cannot modify the access permissions of the `SYSTEM ADMINISTRATORS` group. You can modify the access permissions of only other groups that you assign to the IT resource.

   a. Click **Update Permissions**.

   b. Depending on whether you want to set or remove specific access permissions for groups displayed on this page, select or deselect the corresponding check boxes.

c. Click **Update**.

9. On the Step 3: Set Access Permission to IT Resource page, if you want to unassign a group from the IT resource, then:

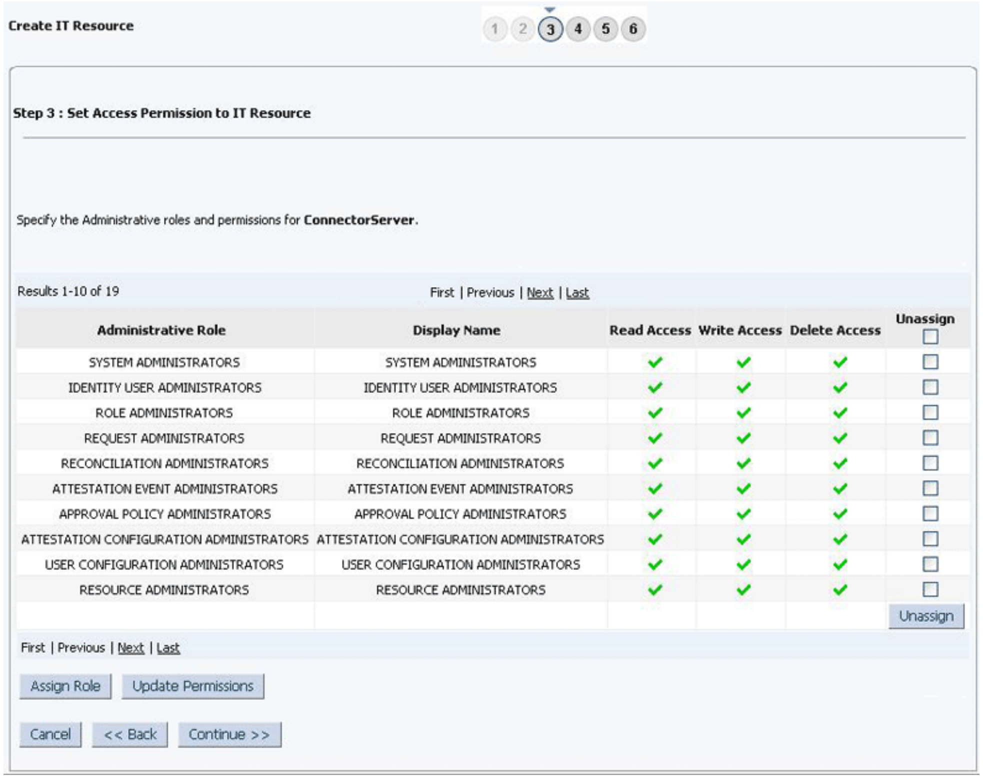
> **Note:**
>
> - This step is optional.
> - You cannot unassign the `SYSTEM ADMINISTRATORS` group. You can unassign only other groups that you assign to the IT resource.

a. Select the **Unassign** check box for the group that you want to unassign.

b. Click **Unassign.**

10. Click **Continue**. Figure 5-3 shows the Step 3: Set Access Permission to IT Resource page.

Figure 5-3 Step 3: Set Access Permission to IT Resource



Create IT Resource

Step 3 : Set Access Permission to IT Resource

Specify the Administrative roles and permissions for **ConnectorServer**.

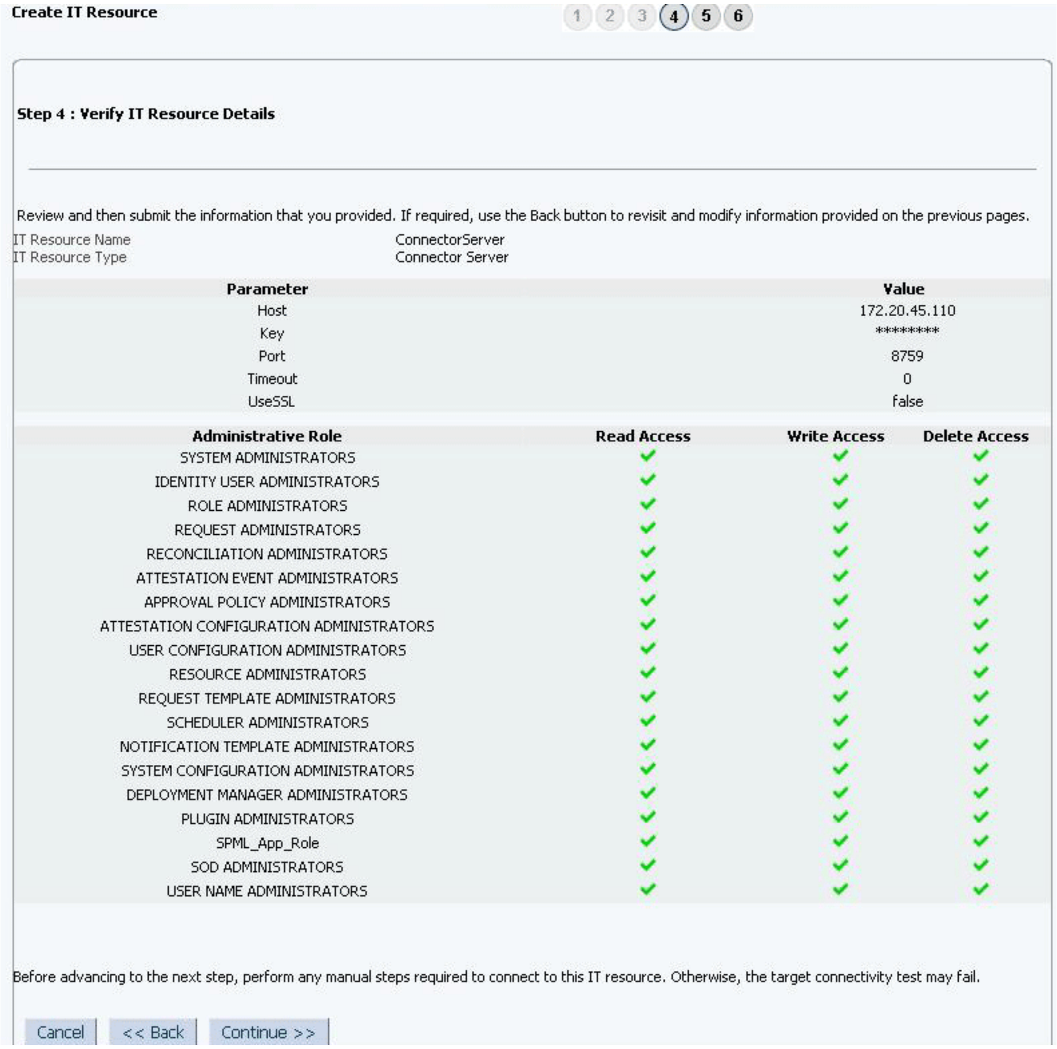
Results 1-10 of 19

Administrative Role	Display Name	Read Access	Write Access	Delete Access	Unassign □
SYSTEM ADMINISTRATORS	SYSTEM ADMINISTRATORS	✓	✓	✓	□
IDENTITY USER ADMINISTRATORS	IDENTITY USER ADMINISTRATORS	✓	✓	✓	□
ROLE ADMINISTRATORS	ROLE ADMINISTRATORS	✓	✓	✓	□
REQUEST ADMINISTRATORS	REQUEST ADMINISTRATORS	✓	✓	✓	□
RECONCILIATION ADMINISTRATORS	RECONCILIATION ADMINISTRATORS	✓	✓	✓	□
ATTESTATION EVENT ADMINISTRATORS	ATTESTATION EVENT ADMINISTRATORS	✓	✓	✓	□
APPROVAL POLICY ADMINISTRATORS	APPROVAL POLICY ADMINISTRATORS	✓	✓	✓	□
ATTESTATION CONFIGURATION ADMINISTRATORS	ATTESTATION CONFIGURATION ADMINISTRATORS	✓	✓	✓	□
USER CONFIGURATION ADMINISTRATORS	USER CONFIGURATION ADMINISTRATORS	✓	✓	✓	□
RESOURCE ADMINISTRATORS	RESOURCE ADMINISTRATORS	✓	✓	✓	□
					Unassign

11. On the Step 4: Verify IT Resource Details page, review the information that you provided on the first, second, and third pages. If you want to make changes in the data entered on any page, click **Back** to revisit the page and then make the required changes.

12. To proceed with the creation of the IT resource, click **Continue**. Figure 5-4 shows Step 4: Verify IT Resource Details page.

Figure 5-4 Step 4: Verify IT Resource Details



Step 4 : Verify IT Resource Details

Review and then submit the information that you provided. If required, use the Back button to revisit and modify information provided on the previous pages.

IT Resource Name	ConnectorServer
IT Resource Type	Connector Server

Parameter	Value
Host	172.20.45.110
Key	********
Port	8759
Timeout	0
UseSSL	false

Administrative Role	Read Access	Write Access	Delete Access
SYSTEM ADMINISTRATORS	✓	✓	✓
IDENTITY USER ADMINISTRATORS	✓	✓	✓
ROLE ADMINISTRATORS	✓	✓	✓
REQUEST ADMINISTRATORS	✓	✓	✓
RECONCILIATION ADMINISTRATORS	✓	✓	✓
ATTESTATION EVENT ADMINISTRATORS	✓	✓	✓
APPROVAL POLICY ADMINISTRATORS	✓	✓	✓
ATTESTATION CONFIGURATION ADMINISTRATORS	✓	✓	✓
USER CONFIGURATION ADMINISTRATORS	✓	✓	✓
RESOURCE ADMINISTRATORS	✓	✓	✓
REQUEST TEMPLATE ADMINISTRATORS	✓	✓	✓
SCHEDULER ADMINISTRATORS	✓	✓	✓
NOTIFICATION TEMPLATE ADMINISTRATORS	✓	✓	✓
SYSTEM CONFIGURATION ADMINISTRATORS	✓	✓	✓
DEPLOYMENT MANAGER ADMINISTRATORS	✓	✓	✓
PLUGIN ADMINISTRATORS	✓	✓	✓
SPML_App_Role	✓	✓	✓
SOD ADMINISTRATORS	✓	✓	✓
USER NAME ADMINISTRATORS	✓	✓	✓

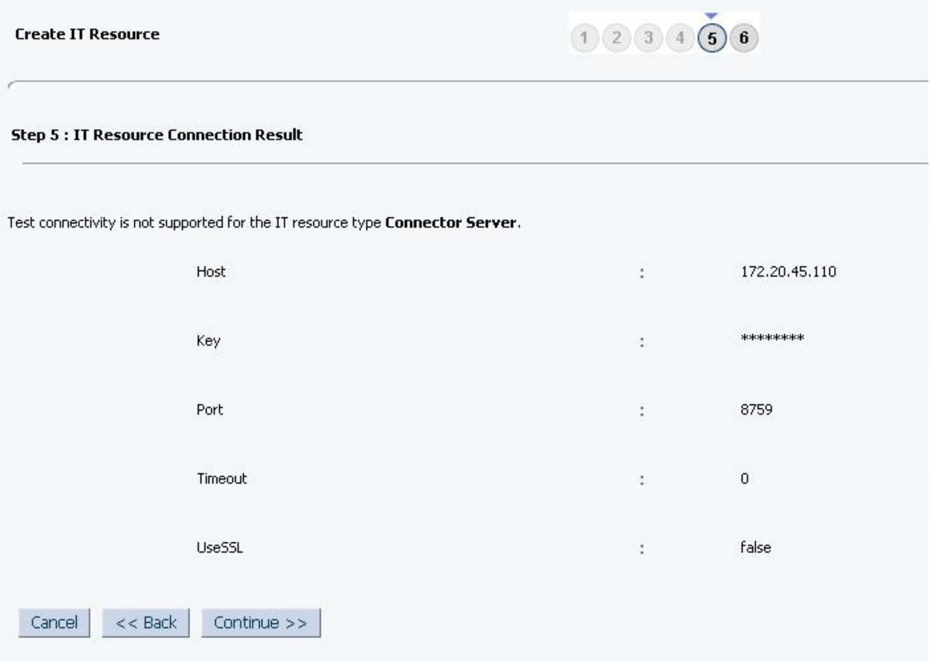
Before advancing to the next step, perform any manual steps required to connect to this IT resource. Otherwise, the target connectivity test may fail.

13. The Step 5: IT Resource Connection Result page displays the results of a connectivity test that is run using the IT resource information. If the test is successful, then click **Continue**. If the test fails, then you can perform one of the following steps:

    - Click **Back** to revisit the previous pages and then make corrections in the IT resource creation information.
    - Click **Cancel** to stop the procedure, and then begin from the first step onward.

    Figure 5-5 shows the Step 5: IT Resource Connection Result page.

Figure 5-5 Step 5: IT Resource Connection Result



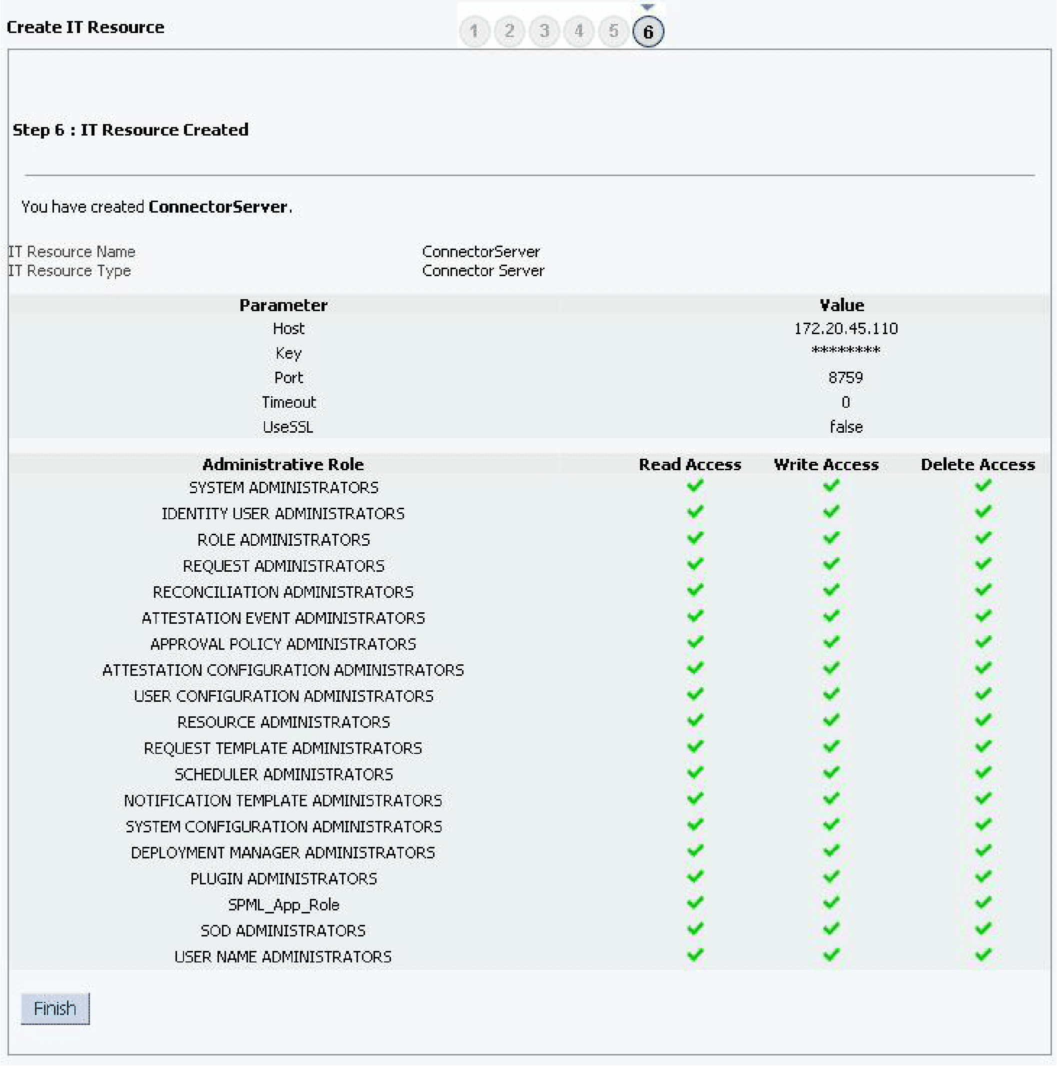
Step 5 : IT Resource Connection Result

Test connectivity is not supported for the IT resource type **Connector Server**.

Host	:	172.20.45.110
Key	:	********
Port	:	8759
Timeout	:	0
UseSSL	:	false

14. Click **Finish**. Figure 5-6 shows the IT Resource Created page.

Figure 5-6 Step 6: IT Resource Created



Step 6 : IT Resource Created

You have created **ConnectorServer**.

IT Resource Name	ConnectorServer
IT Resource Type	Connector Server

Parameter	Value
Host	172.20.45.110
Key	*********
Port	8759
Timeout	0
UseSSL	false

Administrative Role	Read Access	Write Access	Delete Access
SYSTEM ADMINISTRATORS	✓	✓	✓
IDENTITY USER ADMINISTRATORS	✓	✓	✓
ROLE ADMINISTRATORS	✓	✓	✓
REQUEST ADMINISTRATORS	✓	✓	✓
RECONCILIATION ADMINISTRATORS	✓	✓	✓
ATTESTATION EVENT ADMINISTRATORS	✓	✓	✓
APPROVAL POLICY ADMINISTRATORS	✓	✓	✓
ATTESTATION CONFIGURATION ADMINISTRATORS	✓	✓	✓
USER CONFIGURATION ADMINISTRATORS	✓	✓	✓
RESOURCE ADMINISTRATORS	✓	✓	✓
REQUEST TEMPLATE ADMINISTRATORS	✓	✓	✓
SCHEDULER ADMINISTRATORS	✓	✓	✓
NOTIFICATION TEMPLATE ADMINISTRATORS	✓	✓	✓
SYSTEM CONFIGURATION ADMINISTRATORS	✓	✓	✓
DEPLOYMENT MANAGER ADMINISTRATORS	✓	✓	✓
PLUGIN ADMINISTRATORS	✓	✓	✓
SPML_App_Role	✓	✓	✓
SOD ADMINISTRATORS	✓	✓	✓
USER NAME ADMINISTRATORS	✓	✓	✓

6 Using the Flat File Connector

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

- Configuring Reconciliation
- Configuring Reconciliation Jobs
- Uninstalling the Connector

> **Note:**
>
> These sections provide both conceptual and procedural information about configuring the connector. It is recommended that you read the conceptual information before you perform the procedures.

6.1 Configuring Reconciliation

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

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

- Performing Full and Incremental Reconciliation
- Performing Limited Reconciliation

6.1.1 Performing Full and Incremental Reconciliation

The connector supports full as well as incremental reconciliation of users, accounts, and entitlements.

- For users and accounts, any newly added file will be considered as a source for incremental data.
- For deleted users and accounts, if the enterprise target system does not support exporting only the deleted users, then you can use a diff-based approach to reconcile the deleted records into Oracle Identity Governance.
- For entitlements reconciliation, the reconciliation job for loading entitlements can be run in an incremental or full mode. See Providing Settings Information for a Disconnected Resource or Providing Settings Information for a Disconnected Resource for more information about the Flat File Entitlement job for entitlements loading and its parameters.

Full reconciliation involves reconciling all existing user records or accounts from the flat file into Oracle Identity Governance. Incremental reconciliation involves reconciling only user records or accounts that are added or modified after the time-stamp stored in the Latest Token parameter of the reconciliation job.

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 enterprise target system records are reconciled in Oracle Identity Governance.

To perform a full reconciliation run, remove (delete) any value currently assigned to the Latest Token, Incremental Recon Attribute, and Filter parameters of the reconciliation jobs for Accounts Loader for reconciling accounts.

At the end of the reconciliation run, the Latest Token parameter of the reconciliation job for user record or account reconciliation is automatically set to the most recent value obtained from the parameter (for incremental recon) of the flat file. From the next reconciliation run onward, only records created or modified after this most recent value are considered for reconciliation.

6.1.2 Performing Limited Reconciliation

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

The connector supports filters in the reconciliation jobs to fetch those records which match the filter criteria. The filter expression is also passed to custom parsers so that the records can be filtered at the parser level.

You can perform limited reconciliation by creating filters (for example, `startsWith('userName','john')`) for the reconciliation module. This connector provides a Filter parameter (a reconciliation job parameter) that allows you to use any of the Flat File resource attributes to filter the target system records.

See Checking for Filters for information about implementing filters in the custom parser.

For detailed information about ICF Filters, see ICF Filter Syntax of *Developing and Customizing Applications for Oracle Identity Governance*

> **Note:**
>
> You can use the __UID__ attribute name only with the equalTo filter.

6.2 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.

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

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

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

6.3 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 (for example, `ActiveDirectory User; ActiveDirectory Group`) 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*.

7

Extending the Functionality of the Flat File Connector

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

This chapter contains the following optional procedures:

- Configuring Custom Parsers
- Configuring Preprocess and Postprocess Tasks
- Adding New Attributes for Reconciliation
- Configuring Transformation and Validation of Data
- Configuring the Connector for Multiple Installations of the Target System
- Configuring Action Scripts

7.1 Configuring Custom Parsers

Learn about configuring the Flat File connector to process flat files of format other than CSV.

> **Note:**
>
> If you are using the latest version of this connector in the CI-based mode, then see *Oracle Identity Manager Connector Guide for Flat File*, Release 11.1.1 for detailed information on configuring custom parsers. In addition, depending on whether you have configured your target system as a target resource or a trusted source, update the Lookup.FlatFile.Configuration or Lookup.FlatFile.Configuration.Trusted lookup definitions to include a new Code Key entry `headerRowPresent` and set its decode value to `True` if your flat file contains a header row or `False` if it does not contain a header row.

This section contains the following topics:

- Understanding Custom Parser Configuration
- Creating the Custom Parser
- Integrating the Custom Parser with the Flat File Connector
- Creating a CSV File and Updating the Advanced Settings Section

7.1.1 Understanding Custom Parser Configuration

By default, the connector supports processing of flat files exported in the CSV format. To support the processing of flat files exported in formats other than CSV, you must create a custom parser and integrate it with the connector.

Configuring the connector for a custom parser requires you to write a custom parser class that implements the Flat File parser. Then, you must integrate this custom parser class with the connector by creating a JAR to hold the custom parser class and including it in the connector bundle JAR. In addition, in the Advanced Settings section of your application, you must add the **parserClassName** and **customConfigParams** attributes and enter relevant values.

Then, if the exported flat file (containing user data) does not contain a header row, you must add the **headerRowPresent** parameter to the Advanced Settings section and set its value to `False`. In addition, you must create a CSV file containing only the headers and enter the name and complete path of this file in the **flatFileLocation** field of the Advanced Settings section. This is to help the connector understand the schema of the file flat that it needs to parse.

7.1.2 Creating the Custom Parser

Learn about creating a custom parser.

- About Creating the Custom Parser
- Performing Mandatory Attributes Validation
- Checking for Filters
- Handling Multivalued Attributes

7.1.2.1 About Creating the Custom Parser

To configure custom parsers, you must write the code that implements the required custom parser logic in a Java class.

This custom parser class must implement the FlatFileParser interface and the parse method.

> **Note:**
>
> Ensure that the Java version of Oracle Identity Manager and the compiled class is the same.

The following procedure describes the way in which the code must be written to implement the required custom parser logic:

```
class customParser implements FlatFileParser{
void parse(File flatFile, FlatFileRecordHandler recordHandler, ParserConfig
config, boolean isHeaderRowPresent)
{
if isHeaderRowPresent is "true"
{
    void parse(File flatFile, FlatFileRecordHandler recordHandler, ParserConfig
    config) {

    For each record in the flatFile, do the following:

    Start loop

        1) Perform mandatory attribute validation. If all the mandatory attributes
           are present, continue, else skip the record.
           See Performing Mandatory Attributes Validation for more
           information on mandatory attribute validation.

        2) Check for the Filter. Skip this step if the filter is not present. If
           the filter is present and the record satisfies the filter, then
           continue, else, skip the record.
           See Checking for Filters for more information on
           filters.

        3) Create a new FlatFileRecord object and populate all the
           attributes in the record.

            a) Get a list of fields to be sent from the parser, by using
               attributesToGet() method of ParserConfig object.
               The parser should only send these fields back to the requester,
               though the flat file may contain many more.

            b) Check if the field is a single or multivalued field by using the
               isMultiValued(fieldName) method of the FlatFileSchema object that
               is returned by the ParserConfig's getSchema() method.

                i) If the field is single-valued, then add it to the record by
                   using the FlatFileRecord's
                   addSingleValuedAttribute(fieldname,fieldValue).

                ii) If the field is a multivalued Attribute, then check if the
                    attribute is a complex multivalued Attribute. A complex
                    multivalued attribute is an attribute which contains
                    subfields. The getSubFields(fieldname)method of the schema
                    returns the list of subfields if they are present, or returns
                    null, if they are not.

                    ii.a) If the multivalued field does not contain subfields,
                          then add it to the record by using the FlatFileRecord's
                          method addMultiValueAttribute(fieldname, list of
                          attribute values).

                    ii.b) If the multivalued field contains subfields, then the
                          multivalued field is complex field. This value can be
                          added to the record by using the FlatFileRecord's method
                          addComplexAttribute(fieldname, list of map of values).

                          See Handling Multivalued Attributes
                          for more information about multivalued attributes.

        4) Finally, pass the FlatFileRecord back to the requester by calling the
           RecordHandler's handle method.
Endloop


    }
    }
    else
    {
    For each record in the flatFile, do the following:
    Start loop
    1) Create a new FlatFileSchema object and fetch all the fieldNames with
getFieldNames() method of FlatFileSchema.

    2) Create a new FlatFileRecord object and populate all the
           attributes in the record.

            a) Get a list of fields to be sent from the parser, by using
               attributesToGet() method of ParserConfig object.
               The parser should only send these fields back to the requester,
               though the flat file may contain many more.

            b) Check if the field is a single or multivalued field by using the
               isMultiValued(fieldName) method of the FlatFileSchema object that
               is returned by the ParserConfig's getSchema() method.

                i) If the field is single-valued, then add it to the record by
                   using the FlatFileRecord's
                   addSingleValuedAttribute(fieldname,fieldValue).

                ii) If the field is a multivalued Attribute, then check if the
                    attribute is a complex multivalued Attribute. A complex
                    multivalued attribute is an attribute which contains
                    subfields. The getSubFields(fieldname)method of the schema
                    returns the list of subfields if they are present, or returns
                    null, if they are not.

                    ii.a) If the multivalued field does not contain subfields,
                          then add it to the record by using the FlatFileRecord's
                          method addMultiValueAttribute(fieldname, list of
                          attribute values).

                    ii.b) If the multivalued field contains subfields, then the
                          multivalued field is complex field. This value can be
                          added to the record by using the FlatFileRecord's
method
                          addComplexAttribute(fieldname, list of map of values).

                          See Handling Multivalued Attributes
                          for more information about multivalued attributes.

        3)Finally, pass the FlatFileRecord back to the requester by calling the
           RecordHandler's handle method.
Endloop
    }
    }
```

Note:

If you are parsing the file without headers, then the sequence of attributes in the file should be same as that of field names in the Flat File Schema Properties table for your flat file application after parsing the file.

7.1.2.2 Performing Mandatory Attributes Validation

If the mandatory attributes have null or empty values, then you can skip processing such records or log these records at the parser level.

The default CSVParser performs the check on mandatory attributes. If any mandatory attribute contains a value null or empty, the CSV parser creates a directory called "failed" under the directory containing the flat file, and copies the failed records to a flat file with the same name.

The getMandatoryAttrs() method of the FlatFileSchema object returns the list of attributes required by the connector. The FlatFileSchema Object is obtained from the parserConfig parameter of the parse method. The getComplexMandatoryAttributes() method returns the list of complex attributes, and the getSimpleMandatoryAttributes() method returns the list of simple attributes.

7.1.2.3 Checking for Filters

Filters can be specified at the parser level. If a record matches the filter, it is processed, otherwise it can be skipped.

Only simple filters, without the 'and' or 'or' expressions, are supported at the parser level. However, you can specify complex filters by specifying a value for the Filter attribute in the scheduled jobs, as they are supported by ICF. In addition, limited reconciliation comes with a performance overhead, as the entire flat file is parsed to check the filter criteria.

The getFilter() method of the ParserConfig parameter of the parse method returns the FlatFileFilter Object. It is a filter object, represented by the attribute name, the attributes value, and the operator.

A filter such as equalsTo('username','johndoe') can be used in the parser as follows:

```
FlatFileFilter filter=config.getFilter();
String filterFieldname=filter.getFieldName();
String filterValue=filter.getFieldValue();
Operators operator=filter.getOperator();
// since its equalsTO filter, the Operator will be Operators.EQUALS
If(operator.equals(Operators.EQUALS)){
    String userValue=getUserValue(filterFieldname);
    If(userValue.equals(filterValue))
    //process the record
    Else
    //skip the record
}
```

7.1.2.4 Handling Multivalued Attributes

Multivalued attributes are of two types, they are complex and simple. Simple multivalued attributes do not contain any subfields. Complex multivalued attributes contain subfields.

Example 1: Roles is a complex multivalued attribute and it contains subfields like Role Name, Start Date, and End Date. The connector requires the complex data to be represented in a list of mappings that contain subfields and their values in a key-value pair. The following is the format in which the data must be represented:

If there are three roles, such as role1, role2, role3 assigned to the user, then the connector requires a list of these maps in the following format:

List of< Role1 Map, Role2 Map, Role3 Map>.

Here, each role value is in itself a map with key as sub-field name and its value as sub-field value.

Example 2: Groups is a simple multivalued attribute, and it contains such as group1, group2, group3. Here, the connector requires a list of all these values in the following format:

List of<Group1, Group2, Group3>.

7.1.3 Integrating the Custom Parser with the Flat File Connector

Perform this procedure to integrate the custom parser with the connector before it can be used.

1. Create a JAR file to hold the custom parser class.
2. Run the Oracle Identity Manager Download JARs utility to download the org.identityconnectors.flatfile-12.3.0.jar file from the database. This utility is copied into the following location when you install Oracle Identity Governance:

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

   For Microsoft Windows:

   OIM_HOME/server/bin/DownloadJars.bat

   For UNIX:

   OIM_HOME/server/bin/DownloadJars.sh

   When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, URL of the Oracle Identity Governance host computer, context factory value, type of JAR file being downloaded, and the location to which the JAR file is to be downloaded. Specify `4` (ICFBundle) as the value of the JAR type.
3. Update the org.identityconnectors.flatfile-12.3.0.jar file as follows:
   a. Extract the contents of the org.identityconnectors.flatfile-12.3.0.jar file into a temporary directory.
   b. Create a directory called lib in the temporary directory.
   c. Copy the JAR file created in Step 1 into the lib directory.
   d. Re-create the org.identityconnectors.flatfile-12.3.0.jar file by running the following command:

      ```
      jar -cvfm org.identityconnectors.flatfile-1.0.1115.jar META-INF/
      MANIFEST.MF *
      ```

> **Note:**
>
> While re-creating the JAR file, ensure that META-INF\MANIFEST.MF file is unchanged.

4. Run the Oracle Identity Manager Update JARs utility to upload the org.identityconnectors.flatfile-12.3.0.jar file to the database. This utility is copied into the following location when you install Oracle Identity Governance:

   For Microsoft Windows:

   OIM_HOME/server/bin/UpdateJars.bat

   For UNIX:

   OIM_HOME/server/bin/UpdateJars.sh

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

5. Restart the Oracle Identity Governance server after updating the JAR file.

6. Log in to Identity Self Service, and update the Advanced Settings section for your application by adding the parserClassName and customConfigParams attributes. To add these new attributes, click **Add Attribute** and then in the New Attribute window that is displayed, enter values for the following fields and click **OK**:

   - **Name**: Enter `parserClassName` as the name of the attribute being added
   - **Value**: Enter the fully qualified name of the class implementing the custom parser. For example, `com.extension.parser.XMLParser`.
   - **Category**: You need not enter any value for this field. The default value for this field is Custom.
   - **Display Name**: Enter a display name for the attribute being added. For example, `Parser Class Name`.

   Similarly, add the other attribute by entering values for the following fields:

   - **Name**: Enter `customConfigParams` as the name of the attribute being added
   - **Value**:
     Specify the custom configuration parameters used by the custom parser. The value must be in the name-value format as follows:

     NAME1=*VALUE1*;*NAME2*=*VALUE2*

     . For example, `Type=DOM;Version=1.0`.
   - **Category**: You need not enter any value for this field. The default value for this field is Custom.
   - **Display Name**: Enter a display name for the attribute being added. For example, `Custom Config Params`.

7.1.4 Creating a CSV File and Updating the Advanced Settings Section

If your flat file containing user data does not have a header row, then you must specify the same in the Advanced Settings section. In addition, create a CSV file with just the header row and update its location in the Advanced Settings section.

The connector needs to understand the schema of your flat file to successfully perform reconciliation and provisioning operations against it. The presence of a header row in your flat file is important for the connector to obtain the schema of the flat file that it needs to parse.

1. On the computer hosting Oracle Identity Governance, create a CSV file containing only the header row.
2. In the Advanced Settings section of your flat file application:
   a. Click **Add Attribute** to add a new attribute named **headerRowPresent**, and set its value to `False`.
   b. In the flatFileLocation field, enter the complete name and path of the CSV file created in Step 1.
   c. Click **Parse Headers**.

      The header row information is extracted from the CSV file and is listed in the Flat File Schema Properties table.

7.2 Configuring Preprocess and Postprocess Tasks

Learn about preprocess and postprocess tasks and the procedure to integrate them with the connector.

This section contains the following topics:

- Understanding the Preprocess and Postprocess Tasks
- Integrating the Preprocess and Postprocess Tasks with the Flat File Connector

7.2.1 Understanding the Preprocess and Postprocess Tasks

Preprocess and postprocess tasks can be run before and after reconciliation of accounts respectively.

These tasks can be used to perform any job on the flat file directory, like zipping files, unzipping files, encryption and decryption of the complete file dumps or specific fields in the files, virus scan of the files, or any other tasks limited only by the implementation of these tasks.

This section contains the following topics

- About the Preprocess task
- About the Postprocess Task

7.2.1.1 About the Preprocess task

If you are writing code for the preprocess task, then the class must implement the FlatFilePreProcessHandler interface and the preProcess method.

The preProcess method has following parameters:

- flatFileDir

  This parameter specifies the path to the directory containing the flat files.
- configMap

  This parameter contains the mapping of all the configuration parameters from the main configuration lookup definition and the scheduled jobs.

The following procedure describes the way in which the code must be written to implement the FlatFilePreProcessHandler interface:

```
Class preProcessTask implements FlatFilePreProcessHandler{

    void preProcess(java.io.File flatFileDir,
        java.util.Map<java.lang.String,java.lang.Object> configMap) throws
java.lang.Exception{

        //perform the pre process task like unzip and decrypting the files etc.

    }

}
```

7.2.1.2 About the Postprocess Task

If you are writing code for the postprocess task, then the class must implement the FlatFilePostProcessHandler interface and the postProcess method.

The postProcess method has following parameters:

- flatFileDir

  This parameter specifies the path to the directory containing the flat files.
- configMap

  This parameter contains the mapping of all the configuration parameters from the main configuration lookup definition and the scheduled jobs.

The following procedure describes the way in which the code must be written to implement the FlatFilePostProcessHandler interface:

```
Class postProcessTask implements FlatFilePostProcessHandler{

    void postProcess(java.io.File flatFileDir,
        java.util.Map<java.lang.String,java.lang.Object> configMap) throws
java.lang.Exception{

        //perform the post process task like encrypting the files,
        //password protecting files etc.

    }

}
```

7.2.2 Integrating the Preprocess and Postprocess Tasks with the Flat File Connector

This is the procedure to integrate preprocess and postprocess tasks with the connector before they can be used.

To configure preprocess and postprocess tasks:

1. Create a JAR file to hold the preprocess or postprocess task class.
2. Run the Oracle Identity Manager Download JARs utility to download the org.identityconnectors.flatfile-12.3.0.jar file from the 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/DownloadJars.bat
   - **For UNIX:**

     OIM_HOME/server/bin/DownloadJars.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 downloaded, and the location to which the JAR file is to be downloaded. Specify `4` (ICFBundle) as the value of the JAR type.
3. Update the org.identityconnectors.flatfile-12.3.0.jar file as follows:
   a. Extract the contents of the org.identityconnectors.flatfile-12.3.0.jar file into a temporary directory.
   b. Create a directory called lib in the temporary directory.
   c. Copy the JAR file created in Step 1 into the lib directory.
   d. Re-create the org.identityconnectors.flatfile-1.0.1115.jar file by running the following command:

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

      > **Note:**
      >
      > While re-creating the JAR file, ensure that META-INF\MANIFEST.MF file is unchanged.

4. Run the Oracle Identity Manager Update JARs utility to upload the org.identityconnectors.flatfile-12.3.0.jar file to the database. This utility is copied into the following location when you install Oracle Identity Governance:

   - **For Microsoft Windows:**

     OIM_HOME/server/bin/UpdateJars.bat

   - **For UNIX:**

     OIM_HOME/server/bin/UpdateJars.sh

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

5. Restart the Oracle Identity Governance Server after updating the jar.

6. Repeat Steps 1 through 5 to create a JAR file to hold the postprossess task class and copy the JAR file into Oracle Identity Manager.

7. Log in to Identity Self Service, and update the Advanced Settings section for your application by adding the preProcessClassName and postProcessClassName attributes. To add these new attributes, click **Add Attribute** and then in the New Attribute window that is displayed, enter values for the following fields and click **OK**:

   - **Name**: Enter `preProcessClassName` as the name of the attribute being added
   - **Value**: Enter the fully qualified name of the class implementing the preprocess task. For example, `com.extension.parser.PreProcessHandler`.
   - **Category**: You need not enter any value for this field. The default value for this field is Custom.
   - **Display Name**: Enter a display name for the attribute being added. For example, `Preprocess Class Name`.

   Similarly, add the other attribute by entering values for the following fields:

   - **Name**: Enter `postProcessClassName` as the name of the attribute being added
   - **Value**: Enter the fully qualified name of the class implementing the postproces task. For example, `com.extension.parser.PostProcessHandler`.
   - **Category**: You need not enter any value for this field. The default value for this field is Custom.
   - **Display Name**: Enter a display name for the attribute being added. For example, `Postprocess Class Name`.

7.3 Adding New Attributes for Reconciliation

By default, the attributes listed in the Flat File Schema Properties table on the Basic Information page of your application are mapped for reconciliation between the flat file and Oracle Identity Governance. If required, you can add new attributes to the schema for reconciliation.

After creating the application for the flat file, if the exported flat file contains a new attribute, then you can include this new attribute into the schema in one of the following ways:

- In the Advanced Settings section of your application, click **Parse Headers** to fetch all the attributes in the flat file, including the new attribute. Review the Flat File Schema

Properties table and make any changes, if required. For example, change the datatype, mark an attribute an multivalued, and so on. Save the changes to your application.

- In the Flat File Schema Properties table on the Basic Information page of your application, click **Add Field** and fill in the required details to add the new attribute manually. Save the changes to your application.

7.4 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*.

7.5 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*.

7.6 Configuring Action Scripts

For a disconnected resource, you can configure **Action Scripts** by writing your own Groovy scripts while creating the application. For a connected resource, the connector fetches the action scripts that have been defined for the corresponding target application, for example, Salesforce.

You can configure action scripts to run before or after the create, update, 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*.

8 Upgrading the Flat File Connector

If you have already deployed the 11.1.1.5.0 version of this connector, then you can upgrade the connector to the current version 12.2.1.3.0.

> **Note:**
>
> - The connector upgrade from version 11.1.1.5.0 to 12.2.1.3.0 is only supported in the CI-based mode.
> - Before you perform the upgrade, it is strongly recommended that you create a backup of the Oracle Identity Manager database. Refer to the database documentation for information about creating a backup.
> - As a best practice, first perform the upgrade procedure in a test environment.

- Preupgrade Steps
- Upgrade Steps
  Post upgrade Steps

8.1 Preupgrade Steps

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.

Perform the following preupgrade steps:

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

8.2 Upgrade Steps

You must update the decode value for the code key Bundle Version in `the Lookup.<Connector name>.Configuration (Target mode connector) and Lookup.<Connector name>.Configuration.Trusted lookup definition (Trusted mode connector)` from "1.0.1115" to "12.3.0".

8.3 Post upgrade Steps

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

1. Delete the 11g connector bundle jar -- `org.identityconnectors.flatfile-1.0.1115.jar` using DeleteJars utility.
2. Upload the new connector JAR (bundle/org.identityconnectors.flatfile-12.3.0.jar) to the Oracle Identity Manager database as follows:

   Run the Oracle Identity Manager Upload JARs utility to post the new connector bundle to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

   - For Microsoft Windows:

     OIM_HOME/server/bin/UploadJars.bat
   - For UNIX:

     OIM_HOME/server/bin/UploadJars.sh

   > **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.

   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` as the value of the JAR type.
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:
   1. Stop the connector server.
   2. Replace the existing bundle JAR file:
      - `org.identityconnectors.flatfile-1.0.1115.jar` with the new bundle JAR file `org.identityconnectors.flatfile-12.3.0.jar`.
   3. Start the connector server.

9 Troubleshooting the Connector

These are the solutions to problems that you might encounter while using the Flat File connector.

Table 9-1 Troubleshooting for the Flat File Connector

Problem	Reason	Solution

The following error message is encountered:
`org.identityconnectors.framework.common.exceptions.ConnectorException:`
`Error in processing all files in Flat file directory.`
`{filenamexxxx=Mandatory Field Column "xxxxx" not present}. Refer log for more details.`
	The reason for the error could be any one of the following:
	1. The field that is marked as mandatory in the schema file is not present in the header of the flat file.
	2. The header line in the flat file is commented.
	3. The 'fieldDelimiter' specified in the flat file connector configuration is not present in the flat file.
		The following are the solutions to their corresponding causes mentioned in the Reason column of this row:
		1. Ensure that the mandatory fields are present in the header.
		2. Uncomment the header field if commented.
		3. Update the correct field delimiter in the Flat File connector's main configuration lookup definition.

The following error message is encountered:
`org.identityconnectors.framework.common.exceptions.ConfigurationException:`
`Directory does not contain normal files to read xxxx`
	The flat file directory is empty.	Ensure that the directory is valid and contains files to be parsed.

The following error message is encountered:
`org.identityconnectors.framework.common.exceptions.ConnectorException:`
`Specified class not found xxxxxx`
	The reason for the error could be any one of the following:
	1. The value given for the parserClassName, preProcessClassName, or postProcessClassName entries of the main configuration lookup definition is invalid.
	2. The custom class is not available to the class loader.
		The following are the solutions to their corresponding causes mentioned in the Reason column of this row:
		1. Ensure that the fully qualified class name is mentioned for the parserClassName, preProcessClassName, or postProcessClassName entries of the main configuration lookup definition.
		2. Upload the custom parser, preprocess, or postprocess handler JARs to Oracle Identity Manager.

10 Frequently Asked Questions

These are the answers to frequently asked questions related to the functionality of the Flat File connector

1. **While running reconciliation, if new files are added to the flat file directory, will the newly added files be considered by the connector?**

   No. Only files that were present at the beginning of processing will be considered by the connector. The newly added files will be parsed during the next reconciliation run.

2. **Will the Flat File connector parse only the files with a specific extension within the flat file directory?**

   No. The connector does not perform extension checking of files in the flat file directory. It parses all the files in the directory.

11 Known Issues and Workarounds

These are the known issues and workarounds associated with this release of the connector.

- Multicharacter Delimiters Are Not Supported
- Ignore Event API is Not Called
- StartDate and EndDate Values Not Populated in Child Form

11.1 Multicharacter Delimiters Are Not Supported

If a delimiter containing more than one character (for example, $#) is specified in the flat file and in the main configuration lookup definition, then this error is encountered.

```
Only single character delimiters are supported with the exception of "tab" and
"space".
```

This is an issue associated with the connector.

Workaround: The connector supports only single character delimiters. You must either provide a file delimited by a single character or use a preprocess task to parse the file and replace all multicharacter delimiters into a valid character.

11.2 Ignore Event API is Not Called

Suppose a target resource or trusted source reconciliation run is performed to reconcile ten accounts or users from a flat file, the reconciliation run is successful. However, if a target resource or trusted source reconciliation run is performed after copying the same file, then the same number of events is created again, and the Ignore Event API is not called.

This is an issue associated with Oracle Identity Manager.

Workaround:

1. Log in to the Design Console.
2. Expand **Administration,** and then double-click **Lookup Definition.**
3. Search for and open one of the following lookup definitions:
   - If you performed a target resource reconciliation run:

     Lookup.FlatFile.Configuration
   - If you performed a trusted source reconciliation run:

     Lookup.FlatFile.Configuration.Trusted
4. Click **Add.**
5. In the newly added row, enter the following values:
   - **Code Key:** `Ignore Event Disabled`
   - **Decode:** `true`

6. Click **Save.**

11.3 StartDate and EndDate Values Not Populated in Child Form

If a child form contains fields such as Startdate and Enddate, then after a reconciliation run, data for these fields is not displayed in the UI. However, data is populated correctly in the Oracle Identity Manager database.

This is an issue associated with Oracle Identity Manager.

Workaround: Rename the Startdate and Enddate child attributes so as to not match (case insensitive) the parent attribute names. For example, rename the Startdate and Enddate attributes to From and To, respectively.

A

Sample Entries for Users, Currency, Groups, and Roles in a CSV File

These are the sample entries for Users, Currency, Groups, and Roles.

- Sample Entries for Users
- Sample Entries for Currency
- Sample Entries for Groups
- Sample Entries for Roles

A.1 Sample Entries for Users

These are the sample entries for users.

- Sample Entries for Users
- Sample Entries for Accounts with Child Form Data

A.1.1 Sample Entries for Users

These are sample entries for users.

```
UID,UserId,FirstName,LastName,email,Currency,Salary,status,JoiningDate, LastUpdated

"JDOE","John.Doe","John","Doe","john.doe@example.com","USD","12311","Enabled","12202272
00000","1420215120000"

"SDOE","Susan.Doe","Susan","Doe","susan.joe@example.com","INR","54678","Disabled","1220
227200000","1420128720000"

"MJOE","Mikeal.Joe","Mikeal","Joe","mikeal.joe@example.com","EUR","43242","Enabled","12
20227200000","1433171520000"
```

A.1.2 Sample Entries for Accounts with Child Form Data

These are sample entries for accounts with child form data.

```
UID,UserId,FirstName,LastName,email,Currency,Salary,Groups,Roles,status,JoiningDate,Las
tUpdated

"JDOE","John.Doe","John","Doe","john.doe@example.com","USD","12311","Employees
Group;Users Group","User Administrator#1364754600000#1427826600000;Role
Administrator#1220227200000#1220227200000","Enabled","1220227200000","1420215120000"

"SDOE","Susan.Doe","Susan","Doe","susan.joe@example.com","INR","54678","Sports
Group","Advanced User#1220227200000#1220227200000;Group
Administrator#1220227200000#1220227200000","Disabled","1220227200000","1420128720000"

"MJOE","Mikeal.Joe","Mikeal","Joe","mikeal.joe@example.com","EUR","43242","Employees
```

```
Group;Contractors","Service Manager#1364754600000#1427826600000;Advanced
User#1220227200000#1220227200000","Enabled","1220227200000","1433171520000"
```

A.2 Sample Entries for Currency

These are sample entries for currency.

```
currency_code,currency_name

"BSD","Bahamian Dollar"

"CAD","    Canadian Dollar"

"CNY","Yuan Renminbi"

"EUR","Euro"

"HKD","Hong Kong Dollar"

"INR","Indian Rupee"

"USD","US Dollar"
```

A.3 Sample Entries for Groups

These are sample entries for groups.

```
grpId,grpNme

"EMP","Employees Group"

"USR","    Users Group"

"CNTRTS","Contractors"

"SPRTS","Sports Group"

"Analyst","Analyst Group"
```

A.4 Sample Entries for Roles

These are sample entries for roles.

```
roleId,roleName

"ADV","Advanced User"

"Grp_Admin","Group Administrator"

"Role_Admin","Role Administrator"

"SEM","Service Manager"

"SAM","Sales and Marketing Manager"

"User_Admin","User Administrator"
```

B

Files and Directories in the Flat File Connector Installation Package

These are the components of the connector installation media that comprise the Flat File connector.

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

File in the Installation Package Directory	Description
bundle/org.identityconnectors.flatfile-12.3.0.jar	This JAR file is the ICF connector bundle.
configuration/FlatFile-CI.xml	This XML file contains configuration information. The Connector Installer uses this XML file to create connector components that are used for both direct and request-based user account creation.
metadata-generator/bin/FlatFileGenerator.cmd metadata-generator/bin/FlatFileGenerator.sh **Note:** These files are applicable only for a CI-based connector.	This file contains commands to run the metadata generator. Note that the .cmd file is the Microsoft Windows version of the FlatFile Generator. Similarly, the .sh file is the UNIX version of the FlatFile Generator.
metadata-generator/bin/classpath.cmd metadata-generator/bin/classpath-append.cmd **Note:** These files are applicable only for a CI-based connector.	These files contain the commands that add the JAR files (located in the lib directory) to the classpath on Microsoft Windows.
metadata-generator/bin/logging.properties **Note:** This file is applicable only for a CI-based connector.	This file contains the default logging configurations of the metadata generation utility.
metadata-generator/lib/connector-framework.jar **Note:** This file is applicable only for a CI-based connector.	This JAR file contains class files that define the ICF Application Programming Interface (API). This API is used communicate between Oracle Identity Manager and this connector.
metadata-generator/lib/connector-framework-internal.jar **Note:** This file is applicable only for a CI-based connector.	This JAR files contains class files that implement ICF.
metadata-generator/lib/FlatFile-oim-integration.jar **Note:** This file is applicable only for a CI-based connector.	This JAR file contains the class files of the FlatFile generator.
metadata-generator/lib/groovy-all.jar **Note:** This file is applicable only for a CI-based connector.	This JAR file contains the groovy libraries required for running the FlatFile generator.
metadata-generator/resources/FlatFileConfiguration.groovy **Note:** This file is applicable only for a CI-based connector.	This file contains properties that store basic information about the flat file schema, which is used for configuring your flat file either as a trusted source or target resource. In addition, it stores information about the manner in which the connector must connect to the flat file.

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

File in the Installation Package Directory	Description
xml/FlatFile-ConnectorConfig.xml **Note:** This file is applicable only for a CI-based connector.	This XML file contains definitions for the following connector components: • IT resource types • IT resource instance • Lookup definitions • Scheduled jobs
xml/FlatFile-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, correlation rules, and reconciliation jobs.
FlatFile-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, correlation rules, and reconciliation jobs.