

Oracle® Identity Manager Connector Guide for Flat File



Release 11.1.1
E50210-09
October 2020

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

ORACLE®

Oracle Identity Manager Connector Guide for Flat File, Release 11.1.1

E50210-09

Copyright © 2014, 2020, Oracle and/or its affiliates.

Primary Author: Gowri.G.R

Contributors: Jagan Ungati

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

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

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

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

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

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

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

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

Contents

Preface

Audience	x
Documentation Accessibility	x
Related Documents	x
Conventions	x

What's New in the Oracle Identity Manager Connector for Flat File?

Software Updates	xii
Documentation-Specific Updates	xv

1 About the Flat File Connector

1.1	Introduction to the Connector	1-1
1.2	Certified Components	1-1
1.3	Certified Languages	1-2
1.4	Connector Architecture	1-2
1.5	Use Cases Supported by the Connector	1-3
1.5.1	Reconciliation of Records	1-4
1.5.2	Disconnected Resource	1-4
1.5.3	Connected Resource	1-5
1.6	Features of the Connector	1-5
1.6.1	Support for Both Target Resource and Trusted Source Reconciliation	1-6
1.6.2	Full and Incremental Reconciliation	1-6
1.6.3	Limited Reconciliation	1-6
1.6.4	Support for Disconnected Resources	1-6
1.6.5	Resource-Specific Generation of Scheduled Jobs	1-7
1.6.6	Support for Archival	1-7
1.6.7	Support for Custom Parsers	1-7
1.6.8	Support for Reconciling Complex Multivalued Data	1-7
1.6.9	Support for Delimiters	1-8
1.6.10	Support for Comment Characters	1-8
1.6.11	Support for Fault Handling	1-8

1.6.12	Support for Preprocess and Postprocess Handlers	1-8
1.6.13	Support for Reconciliation of Deleted Records	1-8
1.6.14	Support for Transformation and Validation of Account Data	1-9
1.6.15	Support for Connector Server	1-9
1.7	Lookup Definitions Used During Connector Operations	1-9
1.7.1	Lookup.FlatFile.Configuration	1-9
1.7.2	Lookup.FlatFile.Configuration.Trusted	1-10
1.7.3	Lookup.FlatFile.EntFieldMap	1-11
1.7.4	Lookup.FlatFile.UM.Configuration	1-11
1.7.5	Lookup.FlatFile.UM.Configuration.Trusted	1-12

2 Deploying the Flat File Connector

2.1	Preinstallation	2-1
2.1.1	Exporting Flat Files	2-1
2.1.2	Understanding and Creating the Schema File	2-1
2.1.2.1	Understanding the Schema File Format	2-2
2.1.2.2	Configuring the Connector for Status Reconciliation	2-6
2.1.2.3	Configuring the Connector for Date Type	2-6
2.1.2.4	Creating a Schema File	2-6
2.1.3	Installing the Connector for a Connected Resource	2-7
2.1.4	Installing the Connector for a Disconnected resource	2-7
2.1.4.1	Summary of steps to Generate the Connector Components	2-7
2.1.4.2	Preparing the Schema File	2-8
2.1.4.3	Understanding Entries in the FlatFileConfiguration.groovy File	2-8
2.1.4.4	Configuring the FlatFileConfiguration.groovy File	2-15
2.1.4.5	Running the Flat File Generator	2-16
2.1.4.6	Understanding the Generated Connector Package	2-17
2.1.4.7	Installing the Connector Package	2-18
2.1.5	Prerequisites for the Target Resource	2-18
2.1.5.1	Understanding Mapping in the Recon Attribute Map Lookup Definition	2-18
2.1.5.2	Modifying the Reconciliation Field Name and Updating the Reconciliation Field Mapping	2-20
2.1.6	Configuring the Connector for Incremental Reconciliation	2-21
2.1.7	Copying Custom Parser Libraries and Preprocess and Postprocess Handlers	2-21
2.2	Installation	2-22
2.2.1	Installing the Connector on Oracle Identity Manager	2-22
2.2.1.1	Running the Connector Installer	2-22
2.2.1.2	Configuring the IT Resource	2-23
2.2.2	Deploying the Connector in a Connector Server	2-26

2.2.2.1	Installing and Configuring the Connector Server	2-26
2.2.2.2	Running the Connector Server	2-28
2.2.2.3	Installing the Connector on the Connector Server	2-29
2.3	Postinstallation	2-29
2.3.1	Configuring Oracle Identity Manager	2-30
2.3.1.1	Creating and Activating a Sandbox	2-30
2.3.1.2	Creating a New UI Form	2-30
2.3.1.3	Creating an Application Instance	2-31
2.3.1.4	Publishing a Sandbox	2-31
2.3.2	Changing to the Required Input Locale	2-32
2.3.3	Clearing Content Related to Connector Resource Bundles from the Server Cache	2-32
2.3.4	Setting up the Configuration Lookup Definition	2-33
2.3.5	Managing Logging	2-34
2.3.5.1	Understanding Log Levels	2-34
2.3.5.2	Enabling Logging	2-35
2.3.6	Localizing Field Labels in UI Forms	2-37
2.3.7	Configuring the Connector to Ignore Comment Characters	2-38
2.3.8	Creating the IT Resource for the Connector Server	2-39
2.4	Configuring the Connector with the Target Resource	2-45
2.5	Upgrading the Connector	2-47

3 Using the Connector

3.1	Performing First-Time Reconciliation	3-1
3.2	Scheduled Jobs for Lookup Field and Entitlement Synchronization	3-2
3.3	Configuring Reconciliation	3-4
3.3.1	Reconciliation Scheduled Jobs	3-4
3.3.2	Attributes of the Scheduled Jobs	3-5
3.3.2.1	Scheduled Jobs for Reconciliation of User Records	3-6
3.3.2.2	Scheduled Jobs for Reconciliation of Accounts	3-12
3.3.3	Performing Full and Incremental Reconciliation	3-17
3.3.4	Limited Reconciliation	3-18
3.4	Configuring Scheduled Jobs	3-18
3.5	Reconciling Complex Multivalued Data	3-19
3.6	Configuring Fault Handling	3-20
3.7	Configuring Archival	3-20
3.8	Understanding and Configuring Delimiters	3-21
3.8.1	About Delimiters Supported By the Connector	3-21
3.8.2	Configuring Delimiters	3-22

4 Extending the Functionality of the Connector

4.1	Configuring Custom Parsers	4-1
4.1.1	Creating the Custom Parser	4-1
4.1.1.1	About Creating the Custom Parser	4-2
4.1.1.2	About Mandatory Attributes Validation	4-3
4.1.1.3	About Filters	4-3
4.1.1.4	About Handling Multivalued Attributes	4-4
4.1.2	Integrating the Custom Parser with the Flat File Connector	4-4
4.2	Configuring Preprocess and Postprocess Tasks	4-6
4.2.1	Understanding the Preprocess and Postprocess Tasks	4-6
4.2.1.1	About the Preprocess task	4-7
4.2.1.2	About the Postprocess Task	4-7
4.2.2	Integrating the Preprocess and Postprocess Tasks with the Flat File Connector	4-8
4.3	Adding New Attributes for Target Resource Reconciliation	4-10
4.4	Adding New Multivalued Attributes for Target Resource Reconciliation	4-12
4.5	Configuring Validation of Data During Reconciliation	4-14
4.6	Configuring Transformation of Data During Reconciliation	4-16
4.7	Configuring the Connector for Multiple Installations of the Target System	4-18

5 Troubleshooting the Connector

6 Frequently Asked Questions

7 Known Issues and Workarounds

7.1	Multicharacter Delimiters Are Not Supported	7-1
7.2	Ignore Event API is Not Called	7-1
7.3	StartDate and EndDate Values Not Populated in Child Form	7-2
7.4	Error Encountered When Process Recon Event Batch Size is Set to a Smaller Value	7-2

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

A.1	Sample Entries for Users	A-1
A.1.1	Sample Entries for Users	A-1

A.1.2	Sample Entries for Accounts with Child Form Data	A-1
A.2	Sample Entries for Currency	A-2
A.3	Sample Entries for Groups	A-2
A.4	Sample Entries for Roles	A-2

B Sample Schema Files

B.1	Sample Schema File for Users	B-1
B.2	Sample Schema File for Currency	B-2
B.3	Sample Schema File for Groups	B-2
B.4	Sample Schema File for Roles	B-2

C Sample Groovy File

D Files and Directories in the Flat File Installation Media

E Scheduled Jobs for Lookup Fields and Entitlement Synchronization, and Reconciliation

Index

List of Figures

1-1	Architecture of the Connector	1-3
2-1	IT Resources of Type Flat File	2-24
2-2	View IT Resource Details and Parameters page for the Flat File Accounts IT Resource	2-25
2-3	Step 1: Provide IT Resource Information	2-39
2-4	Step 2: Specify IT Resource Parameter Values	2-40
2-5	Step 3: Set Access Permission to IT Resource	2-42
2-6	Step 4: Verify IT Resource Details	2-43
2-7	Step 5: IT Resource Connection Result	2-44
2-8	Step 6: IT Resource Created	2-45
3-1	Job Details Page of the Flat File Entitlements Loader Scheduled Job	3-4
3-2	Job Details Page of the Flat File Users Loader Scheduled Job	3-7
3-3	Sample Multivalued Data Separated by Delimiters	3-22

List of Tables

1-1	Certified Components	1-2
1-2	Entries in the Lookup.FlatFile.Configuration Lookup Definition	1-10
1-3	Entries in the Lookup.FlatFile.Configuration.Trusted Lookup Definition	1-10
1-4	Entries in the Lookup.FlatFile.EntFieldMap Lookup Definition	1-11
1-5	Entries in the Lookup.FlatFile.UM.Configuration Lookup Definition	1-11
1-6	Entries in the Lookup.FlatFile.UM.Configuration.Trusted Lookup Definition	1-12
2-1	Parameters of the IT Resources of Type Flat File for the Enterprise Target System	2-25
2-2	Additional Configuration Entries for the Lookup.FlatFile.Configuration and Lookup.FlatFile.Configuration.Trusted Lookup Definitions	2-33
2-3	Log Levels and ODL Message Type:Level Combinations	2-35
2-4	Parameters of the IT Resource for the Connector Server	2-40
3-1	Attributes of the Scheduled Jobs for Lookup Fields and Entitlement Synchronization	3-3
3-2	Attributes of the Flat File Users Loader and IT_RES_NAME Flat File Users Loader Scheduled Jobs	3-6
3-3	Attributes of the Flat File Users Diff Sync Reconciliation and IT_RES_NAME Flat File Users Delete Diff Reconciliation Scheduled Jobs	3-8
3-4	Attributes of the Flat File Users Delete Reconciliation and IT_RES_NAME Flat File Users Delete Reconciliation Scheduled Jobs	3-10
3-5	Attributes of the Flat File Users Delete Sync Reconciliation and IT_RES_NAME Flat File Users Delete Sync Reconciliation Scheduled Jobs	3-11
3-6	Attributes of the Flat File Accounts Loader and IT_RES_NAME Flat File Accounts Loader Scheduled Jobs	3-12
3-7	Attributes of the Flat File Accounts Diff Sync Reconciliation and IT_RES_NAME Flat File Accounts Delete Diff Reconciliation Scheduled Jobs	3-14
3-8	Attributes of the Flat File Accounts Delete Reconciliation and IT_RES_NAME Flat File Accounts Delete Reconciliation Scheduled Jobs	3-15
3-9	Attributes of the Flat File Accounts Delete Sync Reconciliation and IT_RES_NAME Flat File Accounts Delete Sync Reconciliation Scheduled Jobs	3-16
5-1	Troubleshooting for the Flat File Connector	5-1
D-1	Files and Directories on the Installation Media	D-1
E-1	Scheduled Jobs for Lookup Fields and Entitlement Synchronization, and Reconciliation	E-1

Preface

This guide describes the connector that is used to integrate Oracle Identity Manager with flat files exported from various enterprise target systems.

Audience

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

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

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

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

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

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

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
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 the Oracle Identity Manager Connector for Flat File?

This chapter provides an overview of the updates made to the software and documentation for release 11.1.1.6.0 of the Flat File connector.

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

- [Software Updates](#)

This section describes updates made to the connector software. This section also points out the sections of this guide that have been changed in response to each software update.
- [Documentation-Specific Updates](#)

These include major changes made to this guide. For example, the relocation of a section from the second chapter to the third chapter is a documentation-specific update. These changes are not related to software updates.

Software Updates

The following sections discuss the software updates:

- [Software Updates in Release 11.1.1.6.0](#)
- [Software Updates in Release 11.1.1.5.0](#)

Software Updates in Release 11.1.1.6.0

The metadata generation utility has been revamped in this release. Some of the changes include providing support for disconnected resources, tagging of entitlements, prepopulating all process form fields and so on. The following sections explain each of these software updates in detail:

- [Support for Disconnected Resources](#)
- [Support for Generating Artifacts for All Attributes in the Trusted Source Configuration Mode](#)
- [Automated Tagging of Entitlements](#)
- [Support for Displaying Date Values as Date Picker](#)
- [Resource-Specific Generation of Scheduled Jobs](#)
- [Support for Lookup Fields](#)
- [Generation of Application Instance](#)
- [Support for Prepopulating All Process Form Fields](#)

Support for Disconnected Resources

In the earlier release, the metadata generation utility did not generate Oracle Identity Manager artifacts for disconnected resources. Disconnected resources and mappings between fields in Oracle Identity Manager and corresponding target system attributes had to be manually created.

From this release onward, the metadata generation utility generates 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.

In the FlatFileConfiguration.groovy file, a predefined section named "disconnected" has been added to configure your flat file as a disconnected resource. See [Understanding Entries in the FlatFileConfiguration.groovy File](#) for more information about the entries corresponding to the "disconnected" section.

Support for Generating Artifacts for All Attributes in the Trusted Source Configuration Mode

In the earlier release, if you had configured your flat file as a trusted source, then by default, this connector supported and created mapping for only four fields on the OIM User process form. For all other fields, the corresponding Oracle Identity Manager artifacts for each target system attribute had to be manually created and mapped.

From this release onward, the metadata generation utility generates Oracle Identity Manager artifacts and creates mappings for all target system attributes. Default values for Oracle Identity Manager-specific attributes such as Organization, User Type, and Employee Type are generated too.

Automated Tagging of Entitlements

In the earlier release, entitlement tagging was a manual process. In other words, process form fields to be set up as entitlements had to be manually associated with the corresponding lookup definitions, set the entitlement property to `true`, and then run the Entitlement List and Catalog Synchronization Job scheduled jobs to harvest entitlements and synchronize with the catalog.

From this release onward, the process of entitlement tagging is automated. This is achieved by the addition of the entitlementAttributeList entry to the FlatFileConfiguration.groovy file. Any target system attribute that is listed in the entitlementAttributeList entry is automatically handled as an entitlement. For every attribute listed in the entitlementAttributeList entry, the metadata generator utility creates lookup definitions automatically, assigns them to a process form, and then adds all the required entitlement properties. Entitlements are available in the catalog after the Entitlement Loader scheduled job is run.

See [Understanding Entries in the FlatFileConfiguration.groovy File](#) for more information about the entitlementAttributeList entry such as the format in which values must be specified.

Support for Displaying Date Values as Date Picker

From this release onward, this connector allows you to display any process form field holding a date value as a date picker. The `dateAttributeList` entry has been added to the `FlatFileConfiguration.groovy` file to handle process form fields as date. See [Understanding Entries in the FlatFileConfiguration.groovy File](#) for more information about this entry such as the format in which values must be specified.

Resource-Specific Generation of Scheduled Jobs

In the earlier release, if you created the Flat File connector by using the metadata generation utility, then you had to manually create a copy of the scheduled jobs (for Users, Accounts, or Entitlements) that were shipped along with the ready to use Flat File Connector.

From this release onward, depending on the type of resource (trusted source, target resource, or disconnected resource), the metadata generation utility automatically generates the corresponding scheduled jobs. For example, if you are using the metadata generation utility and configured your flat file as a trusted resource, then the scheduled jobs such as `IT_RES_NAME` Flat File User Loaders, `IT_RES_NAME` Flat File Users Delete Diff Reconciliation, and `IT_RES_NAME` Flat File Users Delete Reconciliation are automatically generated.

See [Reconciliation Scheduled Jobs](#) for information about scheduled jobs that are created when you install the ready to use Flat File connector and a connector created by using the metadata generation utility.

Support for Lookup Fields

In the earlier release, lookup fields in the process form were not automatically generated. You had to manually create the lookup field and flag it as a lookup.

From this release onward, this process has been automated. This is achieved by the addition of the `lookupAttributeList` entry to the `FlatFileConfiguration.groovy` file. For every attribute listed in this entry, the metadata generation utility creates a lookup field and associates it with the corresponding lookup fields on the OIM User process form.

See [Understanding Entries in the FlatFileConfiguration.groovy File](#) for more information about the `lookupAttributeList` entry such as the format in which values must be specified.

Generation of Application Instance

In the earlier release, if you created the connector by using the metadata generation utility, then you had to manually create the application instance, sandbox, and UI forms.

From this release onward, except for the UI form, the metadata generation utility generates the application instance and other required details. You have to manually create the UI form as there are no corresponding APIs in Oracle Identity Manager that automate UI form creation.

Support for Prepopulating All Process Form Fields

In the earlier release, by default, the connector was configured to let Oracle Identity Manager prepopulate only four process form fields during a provisioning operation. To prepopulate the remaining process form fields, you had to manually attach the prepopulate adapter to the field.

From this release onward, you can configure the connector to prepopulate all process form fields. By default, the FlatFileConfiguration.groovy file is configured to prepopulate the User Login, First Name, Last Name, and Password fields. If you want to prepopulate more fields, then you must specify the field names in the prepopulate entry of the FlatFileConfiguration.groovy file.

See [Understanding Entries in the FlatFileConfiguration.groovy File](#) for more information about the prepopulate entry such as the format in which values must be specified.

Software Updates in Release 11.1.1.5.0

This is the first release of the Oracle Identity Manager connector for Flat File. Therefore, there are no software updates in this release.

Documentation-Specific Updates

The following section discusses the documentation-specific updates:

- [Documentation-Specific Updates in Release 11.1.1.6.0](#)
- [Documentation-Specific Updates in Release 11.1.1.5.0](#)

Documentation-Specific Updates in Release 11.1.1.6.0

The following is a documentation-specific update in revision "9" of this guide:

A Note about the transformation jar has been added to [Configuring Transformation of Data During Reconciliation](#).

The following is a documentation-specific update in revision "8" of this guide:

[Understanding Account Qualifiers](#) has been updated to include a new account qualifier named "defaultEnableStatus".

The following is a documentation-specific update in revision "7" of this guide:

The "Oracle Identity Manager" row in [Certified Components](#) has been renamed to "Oracle Identity Governance or Oracle Identity Manager" and also updated for Oracle Identity Governance 12c PS3 (12.2.1.3.0) certification.

The following is a documentation-specific update in revision "6" of this guide:

The "JDK" row has been added to [Table 1-1](#).

The following is a documentation-specific update in revision "5" of this guide:

The "Oracle Identity Manager" row of [Table 1-1](#) has been updated.

The following is a documentation-specific update in revision "4" of this guide:

A "Note" regarding lookup queries has been added at the beginning of [Extending the Functionality of the Connector](#) .

The following are documentation-specific updates in revision "3" of this guide:

- Information regarding configuring generated components has been removed from the "Summary of Steps to Generate the Connector Components" section. You no longer need to configure generated components as they are internally handled by the connector from now onward.
- Section 2.3.1.8, "Configuring the Generated Connector Components" has been removed as the configurations described in this section are internally handled by the connector from now onward.
- Information regarding incremental reconciliation has been added to the "Account Qualifiers" section.
- A note regarding loading date values from a flat file either as Long or String datatype has been added to the "Field Qualifiers" section.
- [Configuring the Connector for Incremental Reconciliation](#) has been added.
- Section 2.6, "Postcloning Steps" has been removed as the connector artifacts can be re-created by using the metadata generation utility, instead of cloning.
- [Configuring the IT Resource](#) and [Reconciliation Scheduled Jobs](#) have been modified.
- Information in [Configuring Reconciliation](#) has been rearranged.
- Information specific to scheduled job attributes has been moved from Section 3.3.3, "Reconciliation Scheduled Jobs" to a new section named [Attributes of the Scheduled Jobs](#).
- Information regarding incremental reconciliation has been modified in [Performing Full and Incremental Reconciliation](#).
- [Table E-1](#) has been modified.
- The following appendixes have been modified:
 - [Sample Entries for Users, Currency, Groups, and Roles in a CSV File](#)
 - [Sample Schema Files](#)
 - [Sample Groovy File](#)

Documentation-Specific Updates in Release 11.1.1.5.0

The following is a documentation-specific update in revision "2" of this release:

Information about limited reconciliation has been modified in [Limited Reconciliation](#).

1

About the Flat File Connector

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

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

- [Introduction to the Connector](#)
- [Certified Components](#)
- [Certified Languages](#)
- [Connector Architecture](#)
- [Use Cases Supported by the Connector](#)
- [Features of the Connector](#)
- [Lookup Definitions Used During Connector Operations](#)

1.1 Introduction to the Connector

Oracle Identity Manager automates access rights management, security, and provisioning of IT resources. Oracle Identity Manager connectors are used to integrate Oracle Identity Manager with external, identity-aware applications. This guide describes the usage of a flat file-based data synchronization approach that you can use to on-board users, entitlements, and entitlement grants from various systems into Oracle Identity Manager by using the Flat File connector.

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 will consume the information in a flat file, thereby enabling the import of this data as Oracle Identity Manager user accounts or entitlements. The flat file connector can be used in a number of situations for offline data loading or when a predefined connector is not available.

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
Oracle Identity Governance or Oracle Identity Manager	<p>You can use one of the following releases of Oracle Identity Governance or Oracle Identity Manager:</p> <ul style="list-style-type: none"> • Oracle Identity Governance 12c PS3 (12.2.1.3.0) • Oracle Identity Manager 11g Release 2 PS2 (11.1.2.2.0) and BP05 in the same release track • Oracle Identity Manager 11g Release 2 BP13 (11.1.2.0.13) • Oracle Identity Manager 11g Release 2 PS1 BP04 (11.1.2.1.4) • Oracle Identity Manager 11g Release 2 PS3 (11.1.2.3.0)
Target System	Any enterprise target system that can export users, accounts, or entitlements to a flat file.
Connector Server	<p>11.1.1.5.0 or later</p> <p>Note: You can download the necessary Java Connector Server from the Oracle Technology Network web page.</p>
Connector Server JDK	JDK 1.6 or later
Flat File format	<p>CSV</p> <p>Note: Formats other than CSV are supported through the use of custom parsers.</p>

1.3 Certified Languages

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

Resource bundles are not part of the connector installation media as the resource bundle entries vary depending on the flat file being used. Field labels in UI forms can be localized, see [Localizing Field Labels in UI Forms](#) for more information.

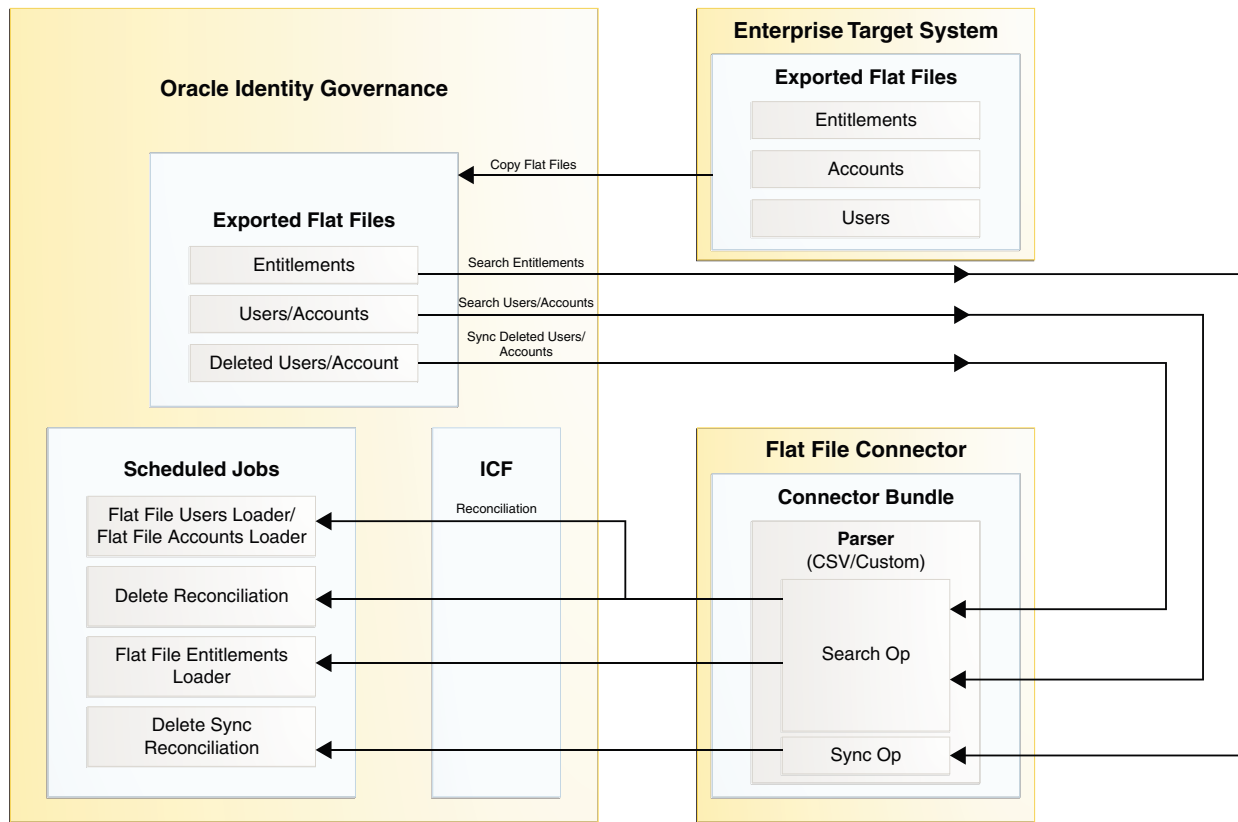
1.4 Connector Architecture

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 installation media contains scheduled jobs that can be used to load users, accounts, and entitlements from a flat file into an existing resource in Oracle Identity Manager.

[Figure 1-1](#) shows the connector integrating the flat files exported from an enterprise target system with Oracle Identity Manager.

Figure 1-1 Architecture of the Connector



The flat files exported from the enterprise target system are stored in a directory that is accessible from Oracle Identity Manager. The connector will sort the files within the directory in an alphanumeric manner, and will process each file based on this order. Additionally, a schema file that describes the attributes in that flat file must be created in the expected format, see [Understanding and Creating the Schema File](#) for more information about the schema file. The location of the directory containing the flat file is specified in the attributes of a scheduled job and the location of the schema file is specified as an IT Resource parameter. When a scheduled job is run, it calls the connector's search implementation, which in turn returns the connector objects to Oracle Identity Manager.

The connector installation media also contains a metadata generator utility to generate Oracle Identity Manager artifacts based on the flat file schema. The metadata generator utility generates a connector package (a zip file) that can be installed in Oracle Identity Manager and used as a resource to load entities by using this connector.

1.5 Use Cases Supported by the Connector

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

- [Reconciliation of Records](#)
- [Disconnected Resource](#)
- [Connected Resource](#)

1.5.1 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.

1.5.2 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.

1.5.3 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 Manager, 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 Manager, 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 Manager 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 Manager.

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 Manager 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 Manager.

1.6 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](#)
- [Resource-Specific Generation of Scheduled Jobs](#)

- [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](#)
- [Support for Reconciliation of Deleted Records](#)
- [Support for Transformation and Validation of Account Data](#)
- [Support for Connector Server](#)

1.6.1 Support for Both Target Resource and Trusted Source Reconciliation

You can configure the exported flat file as a trusted source or target resource for reconciliation of records into Oracle Identity Manager.

See [Attributes of the Scheduled Jobs](#) for more information about the scheduled jobs that are created when you install the connector and their details.

1.6.2 Full and Incremental Reconciliation

After you deploy the connector, you can perform full reconciliation to load all existing user data from the flat file to Oracle Identity Manager.

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.

1.6.3 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 [Limited Reconciliation](#) for more information.

1.6.4 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 [Understanding Entries in the FlatFileConfiguration.groovy File](#).

1.6.5 Resource-Specific Generation of Scheduled Jobs

The connector supports generation of resource-specific scheduled jobs. This means that depending on the type of resource (trusted source, target resource, or disconnected resource), the metadata generation utility automatically generates the corresponding scheduled jobs.

For example, if you are using the metadata generation utility and configured your flat file as a trusted resource, then the scheduled jobs such as *IT_RES_NAME* Flat File User Loaders, *IT_RES_NAME* Flat File Users Delete Diff Reconciliation, and *IT_RES_NAME* Flat File Users Delete Reconciliation are automatically generated.

See [Reconciliation Scheduled Jobs](#) for more information about the scheduled jobs that are created when you install the ready to use Flat File connector and a connector created by using the metadata generation utility.

1.6.6 Support for Archival

The connector supports archival of the processed flat files.

You can specify the archive directory location in the Archive directory attribute while configuring the scheduled jobs, and the connector will move the files from the source directory to the specified location, once all files are processed.

If you do not specify a value for this attribute, then the connector creates an Archived directory within the directory containing the flat file, and saves the processed files in this location.

See [Configuring Archival](#) for more information about archival.

1.6.7 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.

1.6.8 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 [Reconciling Complex Multivalued Data](#) for more information.

1.6.9 Support for Delimiters

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

You can configure delimiters by specifying the values for the `fieldDelimiter`, `multiValueDelimiter`, and `subFieldDelimiter` entries in the `Lookup.FlatFile.Configuration` and `Lookup.FlatFile.Configuration.Trusted` lookup definitions.

See [Understanding and Configuring Delimiters](#) for more information about configuring delimiters.

1.6.10 Support for 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.

This can be configured by specifying the value for the `commentCharacter` entry of the `Lookup.FlatFile.Configuration` lookup definition.

See [Configuring the Connector to Ignore Comment Characters](#) for more information about comment characters.

1.6.11 Support for Fault Handling

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

See [Configuring Fault Handling](#) for more information.

1.6.12 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.

1.6.13 Support for Reconciliation of Deleted Records

You can reconcile data about records that have been deleted on the enterprise target system, by using the exported flat file that has been configured as a trusted source or a target resource.

See [Scheduled Jobs for Reconciliation of User Records](#) and [Scheduled Jobs for Reconciliation of Accounts](#) for more information about the scheduled jobs used for reconciling data about deleted records.

1.6.14 Support for Transformation and Validation of Account Data

You can configure validation of account data that is brought into or sent from Oracle Identity Manager during reconciliation. In addition, you can configure transformation of account data that is brought into Oracle Identity Manager during reconciliation.

The following sections provide more information:

- [Configuring Validation of Data During Reconciliation](#)
- [Configuring Transformation of Data During Reconciliation](#)

1.6.15 Support for Connector Server

Connector Server is a component provided by the Identity Connector Framework (ICF).

By using one or more connector servers, the connector architecture permits your application to communicate with externally deployed bundles. In other words, a connector server enables remote execution of an Oracle Identity Manager connector.

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

1.7 Lookup Definitions Used During Connector Operations

These lookup definitions are automatically created in Oracle Identity Manager after you deploy the connector/

Following are the Lookup definitions used during connector operations:

- [Lookup.FlatFile.Configuration](#)
- [Lookup.FlatFile.Configuration.Trusted](#)
- [Lookup.FlatFile.EntFieldMap](#)
- [Lookup.FlatFile.UM.Configuration](#)
- [Lookup.FlatFile.UM.Configuration.Trusted](#)

1.7.1 Lookup.FlatFile.Configuration

The `Lookup.FlatFile.Configuration` lookup definition holds connector configuration entries that are used during target resource reconciliation operations.

[Table 1-2](#) lists the default entries in this lookup definition.

Table 1-2 Entries in the Lookup.FlatFile.Configuration Lookup Definition

Code Key	Decode	Description
Bundle Name	org.identityconnectors.flatfile	This entry holds the name of the connector bundle package. Do <i>not</i> modify this entry.
Bundle Version	1.0.1115	This entry holds the version of the connector bundle class. Do <i>not</i> modify this entry.
Connector Name	org.identityconnectors.flatfile.FlatFileConnector	This entry holds the name of the connector class. Do <i>not</i> modify this entry.
fieldDelimiter	,	Delimiter of each field in a row.
multiValueDelimiter	;	Delimiter to separate each multivalued data.
subFieldDelimiter	#	Delimiter to separate each subfield within a multivalued field.
textQualifier	"	Character which determines the start and end of text in a value. Any delimiter within the value qualified by the textQualifier will be ignored.
User Configuration Lookup	Lookup.FlatFile.UM.Configuration	This entry holds the name of the lookup definition that contains user-specific configuration properties. Do <i>not</i> modify this entry.

1.7.2 Lookup.FlatFile.Configuration.Trusted

The Lookup.FlatFile.Configuration.Trusted lookup definition holds connector configuration entries that are used during trusted source reconciliation.

[Table 1-3](#) lists the default entries in this lookup definition.

Table 1-3 Entries in the Lookup.FlatFile.Configuration.Trusted Lookup Definition

Code Key	Decode	Description
Bundle Name	org.identityconnectors.flatfile	This entry holds the name of the connector bundle package. Do <i>not</i> modify this entry.
Bundle Version	1.0.1115	This entry holds the version of the connector bundle class. Do <i>not</i> modify this entry.
Connector Name	org.identityconnectors.flatfile.FlatFileConnector	This entry holds the name of the connector class. Do <i>not</i> modify this entry.
fieldDelimiter	,	Delimiter of each field in a row.
multiValueDelimiter	;	Delimiter to separate each multivalued data.
subFieldDelimiter	#	Delimiter to separate each sub-field within a multivalued field.
textQualifier	"	Character which determines the start and end of text in a value. Any delimiter within the value qualified by the textQualifier will be ignored.
User Configuration Lookup	Lookup.FlatFile.UM.Configuration.Trusted	This entry holds the name of the lookup definition that contains user-specific configuration properties. Do <i>not</i> modify this entry.

1.7.3 Lookup.FlatFile.EntFieldMap

The Lookup.FlatFile.EntFieldMap lookup definition is used for populating the code key and decode values of the lookup definition that holds information about target lookups.

You must manually add entries to this lookup definition. To do so, see [Configuring the Connector with the Target Resource](#).

[Table 1-4](#) lists the default entries in this lookup definition.

Table 1-4 Entries in the Lookup.FlatFile.EntFieldMap Lookup Definition

Code Key	Decode
CODE	__NAME__
DECODE	__NAME__

In the Code Key column of this lookup definition CODE and DECODE are the default values. Do not modify these values. By default, CODE and DECODE are mapped to the NameAttribute(__NAME__) mentioned in the schema file.

If you want to map CODE and DECODE to other attributes, modify the values in the Decode column as follows:

In the Decode column of the CODE entry, specify the name of the attribute that you want to populate into the Code Key column of the target lookup definition. Similarly, in the Decode column of the DECODE entry, specify the name of the attribute that you want to populate into the Decode column of the target lookup definition.

1.7.4 Lookup.FlatFile.UM.Configuration

The Lookup.FlatFile.UM.Configuration lookup definition holds configuration entries that are specific to the user object type.

This lookup definition is used during user management operations when your flat file is configured as a target resource. You must manually modify this lookup definition to map the code key to the resource from which accounts must be reconciled.

To modify this lookup definition, see [Configuring the Connector with the Target Resource](#).

[Table 1-5](#) lists the default entries in this lookup definition.

Table 1-5 Entries in the Lookup.FlatFile.UM.Configuration Lookup Definition

Code Key	Decode	Description
Recon Attribute Map	Dummy	This entry holds the name of the lookup definition that maps resource object fields and enterprise target system attributes.

1.7.5 Lookup.FlatFile.UM.Configuration.Trusted

The Lookup.FlatFile.UM.Configuration.Trusted lookup definition holds configuration entries that are specific to the user object type.

This lookup definition is used during user management operations when your flat file is configured as a trusted source. This lookup definition must be modified manually, to map the code key to the resource from which accounts must be reconciled.

To modify this lookup definition, see [Configuring the Connector with the Target Resource](#).

[Table 1-6](#) lists the default entries in this lookup definition.

Table 1-6 Entries in the Lookup.FlatFile.UM.Configuration.Trusted Lookup Definition

Code Key	Decode	Description
Recon Attribute Map	Dummy	This entry holds the name of the lookup definition that maps resource object fields and enterprise target system attributes.

2

Deploying the Flat File Connector

The procedure to deploy the connector can be divided into various stages namely, preinstallation, installation, postinstallation, and upgrading the connector.

- [Preinstallation](#)
- [Installation](#)
- [Postinstallation](#)
- [Configuring the Connector with the Target Resource](#)
- [Upgrading the Connector](#)

2.1 Preinstallation

Preinstallation for the Flat File connector involves performing certain procedures on Oracle Identity Manager and the enterprise target system.

- [Exporting Flat Files](#)
- [Understanding and Creating the Schema File](#)
- [Installing the Connector for a Connected Resource](#)
- [Installing the Connector for a Disconnected resource](#)
- [Prerequisites for the Target Resource](#)
- [Configuring the Connector for Incremental Reconciliation](#)
- [Copying Custom Parser Libraries and Preprocess and Postprocess Handlers](#)

2.1.1 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 Manager.

Ensure to place the exported flat file at a location that is accessible from the computer hosting Oracle Identity Manager.

2.1.2 Understanding and Creating the Schema File

You must create the schema file to let the connector understand the structure of the flat file that the connector must parse.

This section contains the following topics:

- [Understanding the Schema File Format](#)
- [Configuring the Connector for Status Reconciliation](#)
- [Configuring the Connector for Date Type](#)
- [Creating a Schema File](#)

2.1.2.1 Understanding the Schema File Format

The schema file is a .properties file which is used to represent the structure of the flat file that is to be parsed by the connector. It consists of name-value pairs.

It contains details such as datatypes, mandatory attributes, and uid attribute that are specific to the flat file. It is necessary to populate the schema file in the specified format, before using the connector. The schema file is used as an input to the metadata generation utility and is also a mandatory parameter in the IT Resource.

The format in which values must be specified for each property is mentioned in the following topics:

- [Understanding Account Qualifiers](#)
- [Understanding Field Qualifiers](#)

2.1.2.1.1 Understanding Account Qualifiers

These qualifiers are common for the entire flat file and describe certain attributes of an account.

- **FieldNames**

This is a mandatory qualifier. It is a comma separated list of values that must be fetched from the flat file. For example, in a CSV File, this corresponds to the CSV Header line or the name of the column where all the attribute names are present. All child form names, single-valued and multivalued attributes, including the attribute used for performing incremental reconciliation must be specified here. The procedure to configure the schema file for incremental reconciliation is described in [Configuring the Connector for Incremental Reconciliation](#).

The following is a sample value for the FieldNames qualifier:

```
FieldNames=accountID,accountName,firstname,lastname,status,email,roles,privileges,LastUpdated
```

- **UidAttribute**

This is a mandatory qualifier. It refers to the name of the attribute that corresponds to the unique id of the account.

For example: UidAttribute=accountID

- **NameAttribute**

This is a mandatory qualifier. This refers to the name of the attribute that corresponds to a descriptive name of the account.

For example: NameAttribute=accountName

- **PasswordAttribute**

This is an optional qualifier. It refers to the name of the password attribute of the account.

For example: PasswordAttribute=accountPwd

- **StatusAttribute**

This is an optional qualifier. It refers to the attribute which denotes the status of the account.

Oracle Identity Manager requires the values to be true or false. However, if the column in the flat file contains a value other than true or false, then it has to be mapped to `True` or `False`. See the "True" or "False" field qualifiers in [Configuring the Connector for Status Reconciliation](#) for more information.

- **SystemDateFormat**

This is an optional qualifier. If there are date type fields in the flat file and are specified by a date format, then the format string can be specified in the `SystemDateFormat` qualifier. The connector will convert these strings to date according to the format.



See Also:

[Configuring the Connector for Date Type](#)

- **defaultEnableStatus**



Note:

This qualifier is applicable only if you are using the 12.2.1.x version of the Flat File connector in the CI-based mode.

This is a mandatory qualifier if your flat file does not contain an attribute which denotes the status of the account. If you do not include this qualifier, then the connector cannot re-enable a disabled account. Therefore, add this qualifier to set the default status for accounts.

Permitted values for this qualifier are `True` or `False`.

When you set `defaultEnableStatus = True`, the connector sets the status of all new or revoked accounts to `Provisioned` and sets the status of disabled accounts to `Enabled`.

When you set `defaultEnableStatus = False`, the connector sets the status of all new or revoked accounts to `Disabled` and sets the status of enabled accounts to `Disabled`.

2.1.2.1.2 Understanding Field Qualifiers

These qualifiers are specific to each field and are usually specified in one of the following formats:

- The following is the format for parent form fields:

`<FIELDNAME>.<FIELDQUALIFIER>=<VALUE>`

Example: `accountID.Required=true`

- The following is the format for complex child form fields:

`<FIELDNAME>.<SUBFIELDNAME>.<FIELDQUALIFIER>=<VALUE>`.

Example: `Roles.startdate.DataType=Date`

The following are the field qualifiers for which values can be specified:

- **Required**

This field qualifier specifies if the mentioned attribute is mandatory. If the value of this qualifier is set to true, the parser will skip processing the records that do not contain this fieldname.

For example: `accountID.Required=true`

- **Multivalued**

This field qualifier specifies if the mentioned attribute is a multivalued field.

For example: `Roles.Multivalued=true`

- **DataType**

This field qualifier is used to specify the datatype of the field name. The possible values of this qualifier are:

- String
- Long
- Character
- Double
- Float
- Integer
- Boolean
- Byte
- BigDecimal
- BigInteger
- Date

For example: `startDate.DataType=Date`

 **Note:**

You can load date values from a flat file either as Long or String datatype.

- **For loading date values as Long datatype:** Set the `DataType` for the field that must be handled as Date to Long.
- **For loading date values as String datatype:**
 1. From the computer hosting Oracle Identity Manager, obtain the value of the `XL.DefaultDateFormat` system property.
 2. In the schema file, ensure that the datatype of the field that must be handled as Date is set to String and the value of that date field is in String format.
 3. In the flat file, ensure that the string values are in the format obtained in Step 1.

Irrespective of whether you are loading date values as Long or String, you must enter the date field name in the `dateAttributeList` entry of the `FlatFileConfiguration.groovy` file. See the `dateAttributeList` entry in [Configuring the FlatFileConfiguration.groovy File](#) for more information.

- **Subfields**

This field qualifier specifies the subfields in a multivalued attribute if they are present. The subfields must appear in the same order in the flat file as specified here.

For example: `Privileges.Subfields=name,description`

- **EmbeddedObjectClass**

This field qualifier specifies the object class name of child forms that have more than one subfield. The value of this qualifier is used internally by ICF and is mandatory for all complex child forms.

For example: `privileges.EmbeddedObjectClass=Privileges`

- **True**

If any field is marked as Boolean (such as, `Field.DataType=Boolean`), but contains a value other than true or false, the connector can convert the value to a Boolean. True and False qualifiers can take any value which the connector will convert to true and false respectively.

For example: `status.True=Active`

- **False**

If any field is marked as Boolean (such as, `Field.DataType=Boolean`), but contains a value other than true or false, the connector can convert the value to a Boolean. True and False qualifiers can take any value which the connector will convert to true and false respectively.

For example: `status.False=Inactive`

**See Also:**

[Sample Schema Files](#) for a sample ACME schema file

2.1.2.2 Configuring the Connector for Status Reconciliation

Perform this procedure to configure the connector for status reconciliation.

1. Enter the `StatusAttribute` account qualifier and specify the name of the column in the flat file that contains the status of the record as its value.

For example: `StatusAttribute=status`

2. If the column in the flat file contains a value other than true or false, then map this value to the True or False field qualifiers as follows:

- a. For True:

Enter `FIELDNAME.True=STATUSVALUE`

Here, `FIELDNAME` is the name of the column in the flat file that contains the status of the record. `STATUSVALUE` is the value in the flat file which you want to map to the True field qualifier.

For example: `status.True=Enabled`

- b. For False:

Enter `FIELDNAME.False=STATUSVALUE`

Here, `FIELDNAME` is the name of the column in the flat file that contains the status of the record. `STATUSVALUE` is the value in the flat file which you want to map to the False field qualifier.

For example: `status.False=Disabled`

2.1.2.3 Configuring the Connector for Date Type

Perform this procedure to configure the connector for date type.

1. Enter the `DataType` field qualifier and specify its value as `Date`.

For example: `startDate.DataType=Date`

2. Enter the `SystemDateFormat` account qualifier and specify the format string.

For example: `SystemDateFormat=dd-MM-yyyy`

**Note:**

The value of the `SystemDateFormat` account qualifier must be specified according to the Java `SimpleDateFormat` pattern.

2.1.2.4 Creating a Schema File

You must create a schema file describing the structure of your flat file.

1. Create a .properties file.
2. Add entries in the schema file according to requirements of your environment.
The following are the mandatory qualifiers that should be defined in the schema file:
 - FieldNames
 - UidAttribute
 - NameAttribute
3. Provide values for each of the entries that you added. See [Understanding the Schema File Format](#) for more information about the format in which these values must be specified.
4. Save the created .properties file.

2.1.3 Installing the Connector for a Connected Resource

Perform this instruction if you want to load entities into a resource for which a predefined connector is available.

Ensure that you have installed and configured the predefined connector properly.

2.1.4 Installing the Connector for a Disconnected resource

In scenarios where a predefined connector is not available and the Oracle Identity Manager artifacts need to be generated, perform the procedures described here to use the metadata generator utility to generate the connector installer.

This section contains the following topics:

- [Summary of steps to Generate the Connector Components](#)
- [Preparing the Schema File](#)
- [Understanding Entries in the FlatFileConfiguration.groovy File](#)
- [Configuring the FlatFileConfiguration.groovy File](#)
- [Running the Flat File Generator](#)
- [Understanding the Generated Connector Package](#)
- [Installing the Connector Package](#)

2.1.4.1 Summary of steps to Generate the Connector Components

This is a summary of the steps to generate the connector components.

1. Prepare the schema file that needs to be specified in the configuration parameter while configuring the FlatFileConfiguration.groovy file. See [Preparing the Schema File](#) for more information.
2. Specify values for properties in the FlatFileConfiguration.groovy file to configure your flat file either as a trusted source or target resource. See [Configuring the FlatFileConfiguration.groovy File](#) for more information.
3. Run the FlatFile Generator to generate the connector package. See [Running the Flat File Generator](#) for more information.

4. Run the Connector Installer, which uses the XML file (located in the configuration directory of the connector package generated in Step 3) to install the connector that you generated per your requirements. See [Installing the Connector Package](#) for more information.

2.1.4.2 Preparing the Schema File

The schema file is used to represent the structure of the flat file that is to be parsed by the connector.

The schema file is used as an input to the metadata generation utility and is also a mandatory parameter in the IT resource. You must populate the schema file in the specified format, before using the connector. See [Understanding and Creating the Schema File](#) for more information.

2.1.4.3 Understanding Entries in the FlatFileConfiguration.groovy File

The FlatFileConfiguration.groovy file contains sample configurations (one each for trusted source, target resource, and disconnected resource) with default values for most of the entries.

Depending upon your requirements, specify or modify values for entries in this file or create custom configurations according to your requirements. The following are the predefined configurations in the FlatFileConfiguration.groovy file:

- **trusted**
You specify values for the entries in this configuration if you want to configure the flat file as a trusted source.
- **target**
You specify values for the entries in this configuration if you want to configure the flat file as a target resource.
- **disconnected**
You specify values for the entries in this configuration if you want to configure the flat file as a disconnected resource.

The following are the entries in the FlatFileConfiguration.groovy file:

Note:

- Unless specified, all entries described here are common to both trusted source, target resource, and disconnected resource configurations.
- If you do not want to specify a value for any of the optional entries or attributes in the FlatFileConfiguration.groovy file, then comment out that entry or attribute by prefixing it with the double-slash symbol (`//`).

- **itResourceDefName**
This is a mandatory entry. Enter the name of the IT resource type for the enterprise target system. Note that the value that you specify for this entry determines the name of the connector package, connector configuration file, and

connector installer file. For example, if you specify `ACME` as the value of this entry, then the name of the connector package directory is `ACME`. See [Running the Flat File Generator](#) for the directory structure of the connector package.

 **Note:**

It is recommended that the value specified for this entry must be unique for each connector that you create for your flat file, if you plan to install or use the connectors in the same Oracle Identity Manager environment.

In addition, this value will be a part of the names for all connector components (defined in the connector configuration XML file, which is created after you run the Flat File Generator) such as lookup definitions, resource objects, process forms, and scheduled jobs.

For example, if you specify `ACME` as the value of `itResourceDefName` entry, then after you deploy the connector, the configuration lookup definition is created and its name will be `Lookup.ACME.Configuration`.

- `itResourceName`

This is an optional entry. Enter the name of the IT resource for the flat file. By default, the value of this entry is the same as the value of the `itResourceDefName` entry.

Sample value: `FlatFile`

- `trusted`

This is a mandatory entry and present only in the section for trusted source configuration. Set the value of the entry to `true`, if you are configuring your flat file as a trusted source.

- `disconnectedResource`

This is a mandatory entry and present only in the section for disconnected resource configuration. By default, the value of this entry is set to `true`. If you are generating the metadata for a disconnected resource, the do not modify the default value.

If you set the value of this entry to `true`, the utility generates complete metadata for the disconnected resource. If you set the value of this entry to `false`, the utility generates complete metadata for the target or accounts.

- `connectorDir`

This is an optional entry. This entry is the name of the directory that contains the output of the connector. By default, the value of this entry is the same as the value of the `itResourceName` entry.

- `xmlFile`

This is an optional entry. Enter the name and relative path of the XML file that must contain definitions of the connector objects. If you do not specify a value for this entry, then the file name is generated in the following format:

```
IT_RES_DEF_NAME-ConnectorConfig.xml
```

In this format, *IT_RES_DEF_NAME* is the value of the `itResourceDefName` entry.

For example, if you have not specified a value for this entry and FlatFileResource is the value of the itResourceDefName entry, then the name of the XML file that is generated is FlatFileResource-ConnectorConfig.xml.

- configFile

This is an optional entry. Enter the name and relative path of the XML file that contains the configuration information of the connector objects. If you do not specify a value for this entry, then the file name is generated in the following format:

IT_RES_DEF_NAME-CI.xml

In this format, *IT_RES_DEF_NAME* is the value of the itResourceDefName entry.

For example, if you have not specified a value for this entry and FlatFileResource is the value of the itResourceDefName entry, then the name of the XML file that is generated is FlatFileResource-CI.xml.

- propertiesFile

This is an optional entry. Enter the name and relative path of the .properties file which contains the resource bundle translations. If you do not specify a value for this entry, then the file name is generated in the following format:

IT_RES_DEF_NAME-generator.properties

In this format, *IT_RES_DEF_NAME* is the value of the itResourceDefName entry.

For example, if you have not specified a value for this entry and FlatFileResource is the value of the itResourceDefName entry, then the name of the properties file that is generated is FlatFileResource-generator.properties.

- version

This is an optional entry. Enter the release number of the connector.

- bundleJar

This is a mandatory entry. This entry contains the name and relative path of the JAR file containing the ICF bundle that the FlatFile generator will use.

Default value: ../../bundle/org.identityconnectors.flatfile-1.0.1115.jar

Do not change the value of this entry.

- config

This is a mandatory entry in which you specify information about the connector configuration. This connector configuration contains information about the manner in which the connector must behave and connect to the flat file.

This entry contains the schemaFile parameter, which is a mandatory configuration parameter. Enter the name and relative path of the schema file that you want to use. For more information about creating schema files, see [Understanding and Creating the Schema File](#).

- lookupAttributeList

This is an optional entry and is present only in the sections for target resource and disconnected resource configuration. Enter the list of fully qualified attributes or column names in the flat file that must be handled as lookup fields.

The connector creates a lookup field for each of the attributes specified in this entry and associates it with the corresponding lookup fields on the OIM User process form.

If you want to create a lookup field for a single-valued or multivalued field, then enter the value in the following format:

```
['FIELD_NAME']
```

In this format, replace *FIELD_NAME* with the name of the single or multivalued field.

If you want create a lookup field for a multivalued field that is embedded then, enter the value in the following format:

```
['OBJ_CLASS.SUB_FIELD_NAME']
```

In this format, replace:

- *OBJ_CLASS* with the EmbeddedObjectClass name for the child form as specified in the schema file.
- *SUB_FIELD_NAME* with the subfield name for the child form as specified in the schema file.

The default value of this entry is:

```
['ROLES.ROLENAME', 'FirstName']
```

In this value, *ROLES.ROLENAME* is a multivalued field that is embedded. In other words, *ROLES* is the EmbeddedObjectClass name for roles child form as specified in the schema file (that is, *roles.EmbeddedObjectClass=Roles*) and *ROLENAME* is one of the subfields for the roles child form as specified in the schema file (that is *roles.Subfields=ROLENAME*). *FirstName* is a single-valued field.

You can modify the default value to meet the requirements in your environment.

For each of the attributes listed in the *lookupAttributeList* entry, the connector creates a lookup field and scheduled job in the following format:

- Lookup field format:
Lookup.\${*IT_RES_NAME*}.\${*FIELD_NAME*}

- Scheduled job format:
IT_RES_NAME FIELD_NAME Loader

This scheduled job is used to load or reconcile lookup values from the flat file. See [Scheduled Jobs for Lookup Field and Entitlement Synchronization](#) for more information about the list of scheduled jobs for lookup fields and entitlement synchronization and their attributes.

In both the formats, the connector replaces:

- *IT_RES_NAME* with the value of the *itResourceDefName* entry.
- *FIELD_NAME* with the name of the field for which the lookup field is created.

- *entitlementAttributeList*

This is also an optional entry and is present only in the sections for target resource and disconnected resource configuration. Enter the list of fully qualified attributes in the flat file that must be tagged as entitlements.

The connector creates a lookup field for each of the attributes specified in this entry, assigns the lookup fields to a process form, adds all the required properties of entitlements, and then creates a scheduled job in the following format:

```
IT_RES_NAME FIELD_NAME Entitlement Loader
```

This scheduled job synchronizes the catalog with the entitlements automatically. See [Scheduled Jobs for Lookup Field and Entitlement Synchronization](#) for more information about this scheduled job.

If you want to tag entitlements for multivalued fields, then enter the value in the following format:

```
["MULTIVALUED_FIELD_NAME"]
```

If you want to tag entitlements for a multivalued field that is embedded, then enter the value in the following format:

```
["OBJ_CLASS.SUB_FIELD_NAME"]
```

In this format, replace:

- *OBJ_CLASS* with the `EmbeddedObjectClass` name for the child form as specified in the schema file.
- *SUB_FIELD_NAME* with the subfield name for the child form as specified in the schema file.

Default value: ["MyRoles.RoleId", "__GROUPS__"]

You can modify the default value based on your schema.

In this value, `MyRoles.RoleId` is an embedded multivalued field and `__GROUPS__` is a multivalued field.

- **dateAttributeList**

This is an optional entry. Enter the list of attributes that must be handled as date on the process form. Ensure that the data type of the attributes listed here is set to Long in the schema file.

The connector creates a date editor for each of the attributes specified in this entry.

If you want to handle single-valued or multivalued fields as date, then enter the value in the following format:

```
["FIELD_NAME"]
```

In this format, replace *FIELD_NAME* with the name of the single or multivalued field.

If you want to handle an embedded multivalued field as date, then enter the value in the following format:

```
["OBJ_CLASS.SUB_FIELD_NAME"]
```

In this format, replace:

- *OBJ_CLASS* with the `EmbeddedObjectClass` name for the child form as specified in the schema file.
- *SUB_FIELD_NAME* with the subfield name for the child form as specified in the schema file.

Default value: ["JoiningDate"]

You can modify the default value to meet the requirements in your environment.

The following is a sample value for handling embedded multivalued fields as date:

```
[ "MyRole.StartDate", "MyRole.EndDate" ]
```

- **alias**

This is a mandatory entry. The FlatFile generator uses aliases to create relationships between the columns in the flat file and resource object field names in Oracle Identity Manager. In addition, the FlatFile generator uses aliases to shorten long database names to meet the character-length restrictions on form names and form field names in Oracle Identity Manager. Aliasing can be used on column name, form name, and form field name levels. Note that the flat file columns are represented as connector attributes.

Depending on the type of configuration, specify values for one of the following sections:

- For trusted source configuration

In the trusted source configuration section, you use the alias entry to map connector attributes or column names in the flat file to the OIM User form field names. The mappings that you specify here are used to populate entries in the Recon Attribute map lookup definition for trusted source reconciliation.

Note that some of the OIM User form field names do not have the same display name internally. For such fields, you must ensure that you map the connector attribute or column name in the flat file to the internal name rather than the display name. The following table lists the names of the OIM User form display names and their corresponding internal names:

Display Name	Internal Name
Organization	Organization Name
Manager	Manager Login
E-mail	Email

The following is the default value of the alias entry:

```
[ '__NAME__': 'User Login', 'LastName': 'LastName', 'Organization': 'Organization Name', 'Employee Type': 'Xellerate Type', 'Role': 'Role' ]
```

In the default value, note that the "Organization" connector attribute has been mapped to "Organization Name", which is the internal name.

You cannot delete existing mappings in the default value. However, you can modify these mappings.

If you want to add mappings for fields other than the ones already present in the alias entry, then you can add them either to the existing values in the alias entry, or add them to the alias + entry.

The following is the default value of the alias + entry:

```
[ '__ENABLE__': 'Status', 'FirstName': 'First Name', 'email': 'Email', 'JoiningDate': 'Start Date' ]
```

The following is the format in which you must specify values for the alias and alias + entry:

```
[ 'CONN_ATTR1': 'OIM_FIELD1', 'CONN_ATTR2': 'OIM_FIELD2', . . . 'CONN_ATTRn': 'OIM_FIELDn' ]
```

In this format:

- * *CONN_ATTR* is the connector attribute name.

- * *OIM_FIELD* is the name of the field on the OIM User form.
- For target resource or disconnected resource configuration
 - * To map connector attributes or column names in the flat file to fields of the process form. The mappings that you specify here are used to populate entries in the Recon Attribute map and Prov Attribute map lookup definitions for target resource reconciliation.
 - * To set an alias (a unique and shortened name) for the IT resource name specified in the *itResourceName* entry.
 - * To specify a short name for a lengthy process form field name.

When the number of characters in a process form is more than 11, the FlatFile Generator automatically truncates the process form name to 10 characters and then suffixes it with the digit 0. Subsequently, for every process form that results in the same name after truncating, the suffix is incremented by 1. The FlatFile Generator prevents any two process forms from having the same name by using autonumbering. To gain control over the autogenerated form name and to have meaningful form names, you can use an alias to specify a shortened process form name.

This is illustrated by the following example:

Assume that the resource name is FLAT and contains child data that is represented as USER_ROLES in the schema.

When you run the FlatFile Generator, the process form is created and the form name is UD_FLAT_USER_ROLES. As the number of characters in this process form name is more than 11, the FlatFile Generator automatically truncates it to UD_FLAT_US0. The truncated form name, UD_FLAT_US0, is not meaningful.

To avoid encountering such issues or forms with autogenerated names, you can use the alias entry to specify short and meaningful process form names.

The following is the default value of the alias entry in both the target resource and disconnected resource sections:

```
[ '__UID__': 'objectGUID', '__NAME__': 'sAMAccountName' ]
```

You cannot delete existing mappings in the default value as they are mandatory. However, you must modify the default value to match the values of the *UidAttribute* and *NameAttribute* qualifiers in the schema file. For example, in the schema file, if you have set the values of the *UidAttribute* and *NameAttribute* qualifiers to *UID* and *UserId* respectively, then you must set the value of the alias entry to the following:

```
[ '__UID__': 'UID', '__NAME__': 'UserId' ]
```

If you want to add mappings for fields other than the ones already present in the alias entry (in other words, optional aliases), then you can add them either to the existing values in the alias entry, or add them to the alias + entry.

The following is the default value of the alias + entry in both the target resource and disconnected resource sections:

```
[ 'USERROLERELATIONSHIP': 'USRROL', 'comments': 'Description', 'Family
Name': 'Last Name', 'Visibility': 'Status' ]
```

The following is the format in which you must specify values for the alias and alias + entries:

```
[ 'CONN_ATTR1': 'ALIAS_FIELD1', 'CONN_ATTR2': 'ALIAS_FIELD2', . . .
'CONN_ATTRn': 'ALIAS_FIELDn' ]
```

In this format:

- * *CONN_ATTR* is the connector attribute name.
- * *ALIAS_FIELD* is the alias corresponding to the connector attribute or column name in the flat file.

- **prepopulate**

This is an optional entry that is present only in the sections for target resource and disconnected resource. Specify a value for this entry if you want Oracle Identity Manager to prepopulate connector's process form fields from OIM User fields while provisioning a enterprise target system resource.

The default value of this entry is as follows:

```
[ '__NAME__': 'User Login', 'FIRST_NAME': 'First Name', 'LAST_NAME': 'Last
Name', '__PASSWORD__': 'Password' ]
```

This means that the groovy file is configured to prepopulate the following fields by default:

- User Login
- First Name
- Last Name
- Password

You can add fields to or remove fields from the preceding list. The following is the format in which you must specify values for the prepopulate entry:

```
[ 'CONN_ATTR1 or COL_NAME1': 'OIM_FIELD1', 'CONN_ATTR2 or COL_NAME2':
'OIM_FIELD2', . . . 'CONN_ATTRn or COL_NAMEn': 'OIM_FIELDn' ]
```

In this format:

- *CONN_ATTR* is the connector attribute name.
- *COL_NAME* is the column name in the flat file.
- *OIM_FIELD* is the name of the field on the OIM User form.

See *Working with Prepopulate Adapters in Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for more information about attaching and removing prepopulate adapters.

2.1.4.4 Configuring the FlatFileConfiguration.groovy File

Use the FlatFileConfiguration.groovy file to specify the configuration properties of the Oracle Identity Manager metadata to be generated based on the target. In addition, use the FlatFileConfiguration.groovy file to generate the connector package specific to your flat file.

See [Running the Flat File Generator](#) for information about running the Flat File Generator and directory structure of the connector package.

To configure the FlatFileConfiguration.groovy file:

1. In a text editor, open the FlatFileConfiguration.groovy file. This file is located in the FlatFile-*RELEASE_NUMBER*/metadata-generator/resources directory of the installation media.
2. Specify values for entries in one of the following predefined configurations:
 - trusted - for configuring your flat file as a trusted source.
 - target - for configuring your flat file as a target resource.
 - disconnected - for configuring the flat file as a disconnected resource.

 **Note:**

- You can maintain different copies of the configuration groovy file, or add new configurations in the existing FlatFileConfiguration.groovy file.
- The entries in the FlatFileConfiguration.groovy file are described in [Understanding Entries in the FlatFileConfiguration.groovy File](#).

3. Save and close the FlatFileConfiguration.groovy file.

2.1.4.5 Running the Flat File Generator

After configuring the FlatFileConfiguration.groovy file, you must run the Flat File Generator to generate the connector package.

The FlatFile Generator is the FlatFileGenerator.cmd or FlatFileGenerator.sh file that is located in the FlatFile-*RELEASE_NUMBER*/metadata-generator/bin directory of the installation media.

To run the FlatFile Generator:

In a command window, change to the FlatFile-*RELEASE_NUMBER*/metadata-generator/bin directory and run one of the following commands depending on the operating system that you are using:

- **For Microsoft Windows:**

```
FlatFileGenerator.cmd CONFIG_FILE CONFIG_NAME
```

- **For UNIX:**

```
sh FlatFileGenerator.sh CONFIG_FILE CONFIG_NAME
```

In this command, replace:

- *CONFIG_FILE* with the name and absolute or relative path of the FlatFileConfiguration.groovy file.
- *CONFIG_NAME* with the name of the configuration within the FlatFileconfiguration.groovy file, being used for the flat file. The predefined configurations within this file are trusted, target, and disconnected. You can

create additional custom configurations with different names depending on your requirements.

The following is a sample command:

```
FlatFileGenerator.cmd ..\resources\FlatFileConfiguration.groovy target
```

If you encounter any errors while running the Flat File Generator, then you must fix it and then resume running the Flat File Generator.

2.1.4.6 Understanding the Generated Connector Package

The connector package is generated after running the FlatFile Generator.

The connector package is a ZIP file that is generated in the FlatFile-*RELEASE_NUMBER*/metadata-generator/directory. For example, if you have specified ACME App as the value of the itResourceDefName entry in the FlatFileConfiguration.groovy file, then the connector package ZIP (ACME App.zip) file is generated in the FlatFile-11.1.1.6.0/metadata-generator/directory. The directory structure of the connector package is as follows:

```
CONNECTOR_PACKAGE/  
  configuration/  
    IT_RES_DEF-CI.xml  
  resources/  
  xml/  
    IT_RES_DEF-ConnectorConfig.xml
```

In this directory structure:

- *CONNECTOR_PACKAGE* is replaced with the name of the IT resource definition specified as the value of the itResourceDefName entry in the FlatFileConfiguration.groovy file.
- resources/ is an empty directory. Resource bundles can be added to this directory for localization. For information about localizing form fields in UI, see [Localizing Field Labels in UI Forms](#).
- *IT_RES_DEF* is replaced with the name of the IT resource definition specified as the value of the itResourceDefName entry in the FlatFileConfiguration.groovy file.

The *IT_RES_DEF-CI.xml* file that contains configuration information is used by the Connector Installer during installation of the connector for your enterprise target system.

The *IT_RES_DEF-ConnectorConfig.xml* file (also referred to as the connector configuration file) contains definitions for connector components such as IT resource, lookup definitions, scheduled jobs, process forms, and resource objects.

The following behavior is observed after generation of the connector configuration XML file:

The length of a field (column) from the enterprise target system is not fetched into the process form. Therefore, except for the Unique ID and Password fields, the length of all other data fields (of the String data type) on the process form is always set to 255 characters. The length of the Unique ID and Password fields is set to 40 characters.

2.1.4.7 Installing the Connector Package

Before you can start using the connector, you must place the connector package (generated after running the Flat File Generator) in the `OIM_HOME/server/ConnectorDefaultDirectory` directory, unzip the generated connector package, and then run the connector installer for installing the connector package.

To run the connector installer, see [Running the Connector Installer](#). After installing the connector, you must configure the IT resource, see [Configuring the IT Resource](#) for more information.

2.1.5 Prerequisites for the Target Resource

There are certain prerequisites for any target resource for it to work with the Flat File Connector. While most of the connected and disconnected resources already contain these prerequisites, there might be certain cases (such as GTC-based resources) where these steps can apply.

- [Understanding Mapping in the Recon Attribute Map Lookup Definition](#)
- [Modifying the Reconciliation Field Name and Updating the Reconciliation Field Mapping](#)

2.1.5.1 Understanding Mapping in the Recon Attribute Map Lookup Definition

The Recon Attribute Map lookup definition contains the mapping between the resource object reconciliation fields and the target system attributes.

In this connector, the target system attributes correspond to the field (or) column names in the flat file. If a Recon Attribute Map does not exist for the resource, one has to manually create this lookup definition. The Code Key column of this lookup definition must contain the resource object field names and the Decode column must contain the corresponding flat file attributes.

For multivalued fields, the Code Key column of the Recon Attribute Map lookup definition must be in the following format:

MULTIVALUED_RECON_FIELD_NAME~PROPERTY_FIELD_NAME

Sample value 1: lang~Languages

Sample value 2: roles~Rolename

Sample value 3: roles~StartDate

For complex child forms, the corresponding Decode column in the Recon Attribute Map lookup definition must be in the following format:

MULTIVALUED_RECON_FIELD_NAME~EMBEDDED_OBJECT_CLASS_NAME~SUBFIELD_NAME

In this format, replace:

- *MULTIVALUED_RECON_FIELD_NAME* with the name of the multivalued reconciliation field.
- *EMBEDDED_OBJECT_CLASS_NAME* with the EmbeddedObjectClass name for the child form as specified in the schema file.

- *SUBFIELD_NAME* with one of the subfield names for the child form as specified in the schema file.

Sample value: roles~Roles~roleName

In this sample value:

- 'roles' is the multivalued reconciliation field name.
- 'Roles' is the EmbeddedObjectClass name for roles child form as specified in the schema file (that is, roles.EmbeddedObjectClass=Roles).
- 'roleName' is one of the Subfields for roles child form as specified in the schema file (that is, roles.Subfields=roleName,startDate).

The following is a screenshot of the Lookup.ACMEApp.UM.ReconAttrMap lookup definition in which entries for multivalued fields are listed:

Lookup Definition

Code:

Field:

Lookup Type Field Type

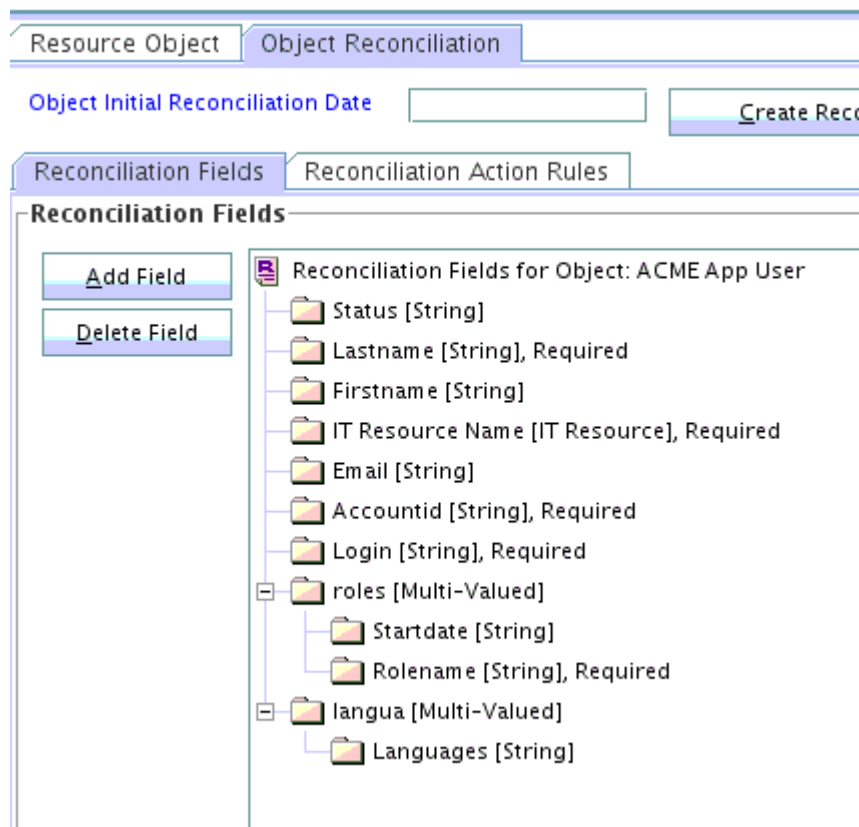
Required:

Group:

Lookup Code Information

	Code Key	Decode ▼
1	Status	__ENABLE__
2	Login	__NAME__
3	Accountid	__UID__
4	Email	email
5	Firstname	firstName
6	langua~Languages	languages
7	Lastname	lastName
8	roles~Rolename	roles~Roles~roleName
9	roles~Startdate	roles~Roles~startDate

The following is a screenshot of the Reconciliation Fields tab for the ACME App User resource object that contains multivalued fields such as roles and langua:



2.1.5.2 Modifying the Reconciliation Field Name and Updating the Reconciliation Field Mapping

The IT Resource Reconciliation field of type 'IT Resource' for the target resource object must be called 'IT Resource Name'. If the name of this Reconciliation field is anything other than 'IT Resource Name', then it must be modified as follows:

1. Log in to the Design Console.
2. Expand **Resource Management**, and click **Resource Objects**.
3. Click on the **Object Reconciliation** tab.
4. Under Reconciliation Fields, search for the field of type IT Resource.
5. If the field name of the reconciliation field is not IT Resource Name, then edit the field to rename the Field Name to IT Resource Name.
6. Check the **Required** check box.
7. Click **Save**.
8. Update the reconciliation field mapping as follows:
 - a. Expand **Process Management**, and click **Process Definition**.
 - b. Search for and open the process definition of the target resource.
 - c. Click the **Reconciliation Field Mappings** tab.
 - d. Edit the IT Resource Field Mapping, and update the Field Name to IT Resource Name.

- e. Click **Save**.

 **Note:**

You must create the Reconciliation Profile after performing the preceding procedure.

2.1.6 Configuring the Connector for Incremental Reconciliation

Perform this procedure to configure your connector for incremental reconciliation.

1. In a text editor, open the schema file created in [Understanding and Creating the Schema File](#).
2. In the FieldNames qualifier, include the name of attribute that holds time stamp information. For example, `LastUpdated`.

This attribute name is specified as the value of the Incremental Recon Attribute of the scheduled job. This scheduled job attribute is explained later in this guide.

3. Set the datatype of the attribute that will be used during incremental reconciliation to `Long`. For example, `LastUpdated.DataType=Long`.
4. In the flat file, ensure that the user or account record contains the incremental value in Long format.

2.1.7 Copying Custom Parser Libraries and Preprocess and Postprocess Handlers

If you are using preprocess or postprocess tasks, or a custom parser through the flat file connector, the Java class files that implement these customizations must be built as a jar and deployed before installing the connector.

See [Configuring Preprocess and Postprocess Tasks](#) and [Configuring Custom Parsers](#) for more information on preprocess and postprocess handlers, and custom parsers.

You must copy the custom jar files as follows:

1. Create a directory named `FlatFile-RELEASE_NUMBER` under the following directory:

`OIM_HOME/server/ConnectorDefaultDirectory/targetsystems-lib/`

For example, if you are using release 11.1.1.6.0 of this connector, then create a directory named `FlatFile-11.1.1.6.0` in the `OIM_HOME/server/ConnectorDefaultDirectory/targetsystems-lib/`.

2. Copy the custom jar files created in [Integrating the Custom Parser with the Flat File Connector](#) and [Integrating the Preprocess and Postprocess Tasks with the Flat File Connector](#) to the `OIM_HOME/server/ConnectorDefaultDirectory/targetsystems-lib/FlatFile-RELEASE_NUMBER`.

For example, if you have built a jar in the name `xml-parser.jar`, it has to be copied to `OIM_HOME/server/ConnectorDefaultDirectory/targetsystems-lib/FlatFile-RELEASE_NUMBER` directory.

2.2 Installation

Depending on where you want to run the connector code (bundle), the connector provides these installation options.

- To run the connector code locally in Oracle Identity Manager, perform the procedure described in [Installing the Connector on Oracle Identity Manager](#).
- To run the connector code remotely in a Connector Server, perform the procedures described in [Installing the Connector on Oracle Identity Manager](#) and [Deploying the Connector in a Connector Server](#).

2.2.1 Installing the Connector on Oracle Identity Manager

Installing the connector involves performing these procedures.

- [Running the Connector Installer](#)
- [Configuring the IT Resource](#)

2.2.1.1 Running the Connector Installer

To run the Connector Installer:

1. Copy the contents of the connector installation media directory into the following directory:
OIM_HOME/server/ConnectorDefaultDirectory
2. If you have not already done so, create a directory in *OIM_HOME/server/ConnectorDefaultDirectory/targetsystems-lib* with the same name as the connector package. For the flat file connector, this name is FlatFile-11.1.1.6.0. For example:
OIM_HOME/server/ConnectorDefaultDirectory/targetsystems-lib/FlatFile-11.1.1.6.0
Copy the external JAR files to this directory. See [Copying Custom Parser Libraries and Preprocess and Postprocess Handlers](#) for more information.
3. Log in to Oracle Identity System Administration.
4. In the left pane, under System Management, click **Manage Connector**.
5. In the Manage Connector page, click **Install**.
6. From the Connector List list, select **FlatFile Connector RELEASE_NUMBER**. This list displays the names and release numbers of connectors whose installation files you copy into the default connector installation directory in Step 1.
If you have copied the installation files into a different directory, then:
 - a. In the **Alternative Directory** field, enter the full path and name of that directory.
 - b. To repopulate the list of connectors in the Connector List list, click **Refresh**.
 - c. From the Connector List list, select **FlatFile Connector RELEASE_NUMBER**.
7. Click **Load**.
8. To start the installation process, click **Continue**.

The following tasks are performed in sequence:

- a. Configuration of connector libraries
- b. Import of the connector XML files (by using the Deployment Manager)
- c. Compilation of adapters

On successful completion of a task, a check mark is displayed for the task. If a task fails, then an X mark and a message stating the reason for failure are displayed. Depending on the reason for the failure, make the required correction and then perform one of the following steps:

- Retry the installation by clicking **Retry**.
 - Cancel the installation and begin again from Step 1.
9. If all three tasks of the connector installation process are successful, then a message indicating successful installation is displayed. In addition, a list of the steps that you must perform after the installation is displayed. These steps are as follows:
 - a. Ensuring that the prerequisites for using the connector are addressed

 **Note:**

At this stage, run the Oracle Identity Manager PurgeCache utility to load the server cache with content from the connector resource bundle in order to view the list of prerequisites. See [Clearing Content Related to Connector Resource Bundles from the Server Cache](#) for information about running the PurgeCache utility.

There are no prerequisites for some predefined connectors.

- b. Configuring the IT resource for the connector

Record the name of the IT resource displayed on this page. The procedure to configure the IT resource is described later in this guide.

- c. Configuring the scheduled jobs that are created when you installed the connector.

Record the names of the scheduled jobs displayed on this page. The procedure to configure these scheduled jobs is described later in this guide.

When you run the Connector Installer, it copies the connector files to destination directories on the Oracle Identity Manager host computer. These files are listed in [Table D-1](#).

2.2.1.2 Configuring the IT Resource

Depending on whether you have installed only the ready-to-use Flat File connector that is shipped with the connector installation media, or created the connector using the metadata generation utility, a set of IT resources are created.

If you have installed only the ready-to-use Flat File connector, then the following IT resources of type "Flat File" are available when the connector is installed:

- Flat File Entitlements

Configure this IT resource when you want to load lookups and entitlements from the flat file.

- Flat File Users

Configure this IT resource when you want to load users from the flat file and you have configured your enterprise target system as a trusted source.

- Flat File Accounts

Configure this IT resource when you want to load accounts from the flat file and you have configured your enterprise target system as a target resource.

If you have created the Flat File connector using the metadata generation utility, then depending on whether you have configured the flat file as a trusted source, target resource, or disconnected resource, the following IT resources of the type *IT_RES_NAME* FlatFile are available:

- For trusted source configuration:
 - IT_RES_NAME* Users Configure this IT resource when you want to load users from the flat file.
- For target resource or disconnected resource configuration:
 - *IT_RES_NAME* Accounts Configure this IT resource when you want to load accounts from the flat file.
 - *IT_RES_NAME* FieldName Configure this IT resource when you want to load lookups and entitlements from the flat file.

]

Figure 2-1 shows the IT resources of type Flat File when the ready-to-use Flat File connector is installed.

Figure 2-1 IT Resources of Type Flat File

IT Resource Type	
Flat File	
<input type="button" value="Search"/> <input type="button" value="Create IT Resource"/> <input type="button" value="Clear"/>	
Results 1-3 of 3	First Previous Next Last
IT Resource Name	IT Resource Type
Flat File Entitlements	Flat File
Flat File Users	Flat File
Flat File Accounts	Flat File

Depending on the IT Resource that you want to configure, you must specify values for the parameters of the IT resources as follows:

1. Log in to Oracle Identity System Administration.
2. In the left pane, under Configuration, click **IT Resource**.

3. In the IT Resource Name field on the Manage IT Resource page, enter the IT resource name (for example, Flat File Entitlements) and then click **Search**. Alternatively, from the IT Resource Type menu, select the name of the IT resource (for example, **Flat File Entitlements**), and then click **Search**.
4. Click the Edit icon for the IT resource.
5. From the list at the top of the page, select **Details and Parameters**.
6. Specify values for the parameters of the IT resource (for example, Flat File Entitlements). Note that the parameters for all IT resources are the same. [Table 2-1](#) describes each parameter of all the IT Resources.

Table 2-1 Parameters of the IT Resources of Type Flat File for the Enterprise Target System

Parameter	Description
Configuration Lookup	Name of the lookup definition that contains the configuration information used during reconciliation. If you have configured your enterprise target system as a target resource, then enter <code>Lookup.FlatFile.Configuration</code> . If you have configured your enterprise target system as a trusted source, then enter <code>Lookup.FlatFile.Configuration.Trusted</code> . Default value: <code>Lookup.FlatFile.Configuration</code>
Connector Server Name	Name of the IT resource of the type "Connector Server." You create an IT resource for the Connector Server in Creating the IT Resource for the Connector Server . Note: Enter a value for this parameter only if you have deployed the Flat File connector in the Connector Server. Sample value: Flat File Connector Server
schemaFile	Absolute path of the schema file. Sample value: <code>D:\flatfile\schema.properties</code>

[Figure 2-2](#) shows the View IT Resource Details and Parameters page.

Figure 2-2 View IT Resource Details and Parameters page for the Flat File Accounts IT Resource

View IT Resource Details and Parameters

You can view additional information about this IT resource : Details and Parameters ▾

IT Resource Name Flat File Accounts
IT Resource Type Flat File

Parameter	Value
Configuration Lookup	Lookup.FlatFile.Configuration
Connector Server Name	
schemaFile	/home/jim/files/schema/acme.properties

Edit

[Back to Search Results](#)

7. To save the values, click **Update**.

2.2.2 Deploying the Connector in a Connector Server

You can deploy the Flat File connector either locally in Oracle Identity Manager or remotely in the Connector Server.

A **connector server** is an application that enables remote execution of an Identity Connector, such as the Flat File connector.

Note:

- To deploy the connector bundle remotely in a Connector Server, you must first deploy the connector in Oracle Identity Manager, as described in [Installing the Connector on Oracle Identity Manager](#).
- See [Creating the IT Resource for the Connector Server](#) for related information.

This procedure can be divided into the following stages:

- [Installing and Configuring the Connector Server](#)
- [Running the Connector Server](#)
- [Installing the Connector on the Connector Server](#)

2.2.2.1 Installing and Configuring the Connector Server

Connector servers are available in two implementations:

- As a .Net implementation that is used by Identity Connectors implemented in .Net
- As a Java Connector Server implementation that is used by Java-based Identity Connectors

The Flat File connector is implemented in Java, so you can deploy this connector to a Java Connector Server.

Use the following steps to install and configure the Java Connector Server:

Note:

Before you deploy the Java Connector Server, ensure that you install the JDK or JRE on the same computer where you are installing the Java Connector Server and that your `JAVA_HOME` or `JRE_HOME` environment variable points to this installation.

1. Create a new directory on the computer where you want to install the Java Connector Server.

 **Note:**

In this guide, `CONNECTOR_SERVER_HOME` represents this directory.

2. Unzip the Java Connector Server package in the new directory created in Step 1. You can download the Java Connector Server package from the Oracle Technology Network.
3. Open the `ConnectorServer.properties` file located in the `conf` directory. In the `ConnectorServer.properties` file, set the following properties, as required by your deployment.

Property	Description
<code>connectorserver.port</code>	Port on which the Java Connector Server listens for requests. Default is <code>8759</code> .
<code>connectorserver.bundleDir</code>	Directory where the connector bundles are deployed. Default is <code>bundles</code> .
<code>connectorserver.libDir</code>	Directory in which to place dependent libraries. Default is <code>lib</code> .
<code>connectorserver.usessl</code>	If set to <code>true</code> , the Java Connector Server uses SSL for secure communication. Default is <code>false</code> . If you specify <code>true</code> , use the following options on the command line when you start the Java Connector Server: <ul style="list-style-type: none"> • <code>-Djavax.net.ssl.keyStore</code> • <code>-Djavax.net.ssl.keyStoreType (optional)</code> • <code>-Djavax.net.ssl.keyStorePassword</code>
<code>connectorserver.ifaddress</code>	Bind address. To set this property, uncomment it in the file (if necessary). The bind address can be useful if there are more NICs installed on the computer.
<code>connectorserver.key</code>	Java Connector Server key.

4. Set the properties in the `ConnectorServer.properties` file, as follows:
 - To set the `connectorserver.key`, run the Java Connector Server with the `/setKey` option.

 **Note:**

For more information, see [Running the Connector Server](#).

- For all other properties, edit the `ConnectorServer.properties` file manually.
5. The `conf` directory also contains the `logging.properties` file, which you can edit if required by your deployment.

**Note:**

Oracle Identity Manager has no built-in support for connector servers, so you cannot test your configuration.

2.2.2.2 Running the Connector Server

To run the Java Connector Server, use the ConnectorServer.bat script for Windows and use the ConnectorServer.sh script for UNIX as follows:

1. Make sure that you have set the properties required by your deployment in the ConnectorServer.properties file, as described in [Installing and Configuring the Connector Server](#).
2. Change to the `CONNECTOR_SERVER_HOME\bin` directory and find the ConnectorServer.bat script.

The ConnectorServer.bat supports the following options:

Option	Description
<code>/install [serviceName]</code> <code>["-J java-option"]</code>	Installs the Java Connector Server as a Windows service. Optionally, you can specify a service name and Java options. If you do not specify a service name, the default name is ConnectorServerJava.
<code>/run ["-J java-option"]</code>	Runs the Java Connector Server from the console. Optionally, you can specify Java options. For example, to run the Java Connector Server with SSL: <code>ConnectorServer.bat /run "-J-Djavax.net.ssl.keyStore=mykeystore.jks" "-J-Djavax.net.ssl.keyStorePassword=password"</code>
<code>/setKey [key]</code>	Sets the Java Connector Server key. The ConnectorServer.bat script stores the hashed value of the key in the connectorserver.key property in the ConnectorServer.properties file.
<code>/uninstall [serviceName]</code>	Uninstalls the Java Connector Server. If you do not specify a service name, the script uninstalls the ConnectorServerJava service.

3. If you need to stop the Java Connector Server, stop the respective Windows service.

2.2.2.3 Installing the Connector on the Connector Server

 **See Also:**

Using an Identity Connector Server in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about installing and configuring connector server and running the connector server

If you need to deploy the Flat File connector into the Java Connector Server, then follow these steps:

1. Stop the Java Connector Server.

 **Note:**

Ensure that you are using latest framework JARs of Oracle Identity Manager to keep the Connector Server consistent with your Oracle Identity Manager instance. To do so:

Copy the framework JAR files, `connector-framework.jar` and `connector-framework-internal.jar`, from the `OIM_HOME/server/ext/internal` directory to the `CONNECTOR_SERVER_HOME/lib/framework` directory.

2. Copy the connector bundle JAR file (`org.identityconnectors.flatfile-1.0.1115.jar`) from the installation media into the Java Connector Server `CONNECTOR_SERVER_HOME/bundles` directory.
3. If you are using custom jars for parsers, preprocess and post process handlers, then copy the necessary jars to `CONNECTOR_SERVER_HOME/lib` directory.
4. Start the Java Connector Server.

2.3 Postinstallation

Configuring the Oracle Identity Manager server involves performing procedures such as configuring Oracle Identity Manager, setting up Configuration lookup definitions, enabling logging and so on.

- [Configuring Oracle Identity Manager](#)
- [Changing to the Required Input Locale](#)
- [Clearing Content Related to Connector Resource Bundles from the Server Cache](#)
- [Setting up the Configuration Lookup Definition](#)
- [Managing Logging](#)
- [Localizing Field Labels in UI Forms](#)
- [Configuring the Connector to Ignore Comment Characters](#)
- [Creating the IT Resource for the Connector Server](#)

2.3.1 Configuring Oracle Identity Manager

You must create additional metadata such as a UI form and an application instance for the resource into which you want to load data by using the flat file connector.

These procedures are described in the following topics:

- [Creating and Activating a Sandbox](#)
- [Creating a New UI Form](#)
- [Creating an Application Instance](#)
- [Publishing a Sandbox](#)

2.3.1.1 Creating and Activating a Sandbox

Create and activate a sandbox as follows. For detailed instructions, see *Managing Sandboxes in Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

1. Log in to Oracle Identity System Administration.
2. In the upper right corner of the page, click the **Sandboxes** link.
The Manage Sandboxes page is displayed.
3. On the toolbar, click **Create Sandbox**.
4. In the Create Sandbox dialog box, enter values for the following fields:
 - **Sandbox Name:** Enter a name for the sandbox.
 - **Sandbox Description:** Enter a description of the sandbox.
5. Click **Save and Close**.
6. Click **OK** on the confirmation message that is displayed.
The sandbox is created and displayed in the Available Sandboxes section of the Manage Sandboxes page.
7. From the table showing the available sandboxes in the Manage Sandboxes page, select the newly created sandbox that you want to activate.
8. On the toolbar, click **Activate Sandbox**.
The sandbox is activated.

2.3.1.2 Creating a New UI Form

Create a new UI form as follows. For detailed instructions, see *Managing Forms in Oracle Fusion Middleware Administering Oracle Identity Manager*.

1. In the left pane, under Configuration, click **Form Designer**. The Form Designer page is displayed.
2. From the Actions menu, select **Create**. Alternatively, click **Create** on the toolbar. The Create Form page is displayed.
3. On the Create Form page, enter values for the following UI fields:

- **Resource Type:** Select the resource object that you want to associate the form with. For example, **ACME User**.
 - **Form Name:** Enter a name for the form.
4. Click **Create**.
A message is displayed stating that the form is created.

2.3.1.3 Creating an Application Instance

Create an application instance as follows. For detailed instructions, see *Managing Application Instances* in *Oracle Fusion Middleware Administering Oracle Identity Manager*.

1. In the left pane of the System Administration console, under Configuration, click **Application Instances**. The Application Instances page is displayed.
2. From the Actions menu, select **Create**. Alternatively, click **Create** on the toolbar. The Create Application Instance page is displayed.
3. Specify values for the following fields:
 - **Name:** The name of the application instance.
 - **Display Name:** The display name of the application instance.
 - **Description:** A description of the application instance.
 - **Resource Object:** The resource object name. Click the search icon next to this field to search for and select **ACME User**.
 - **IT Resource Instance:** The IT resource instance name. Click the search icon next to this field to search for and select **ACME Application**.
 - **Form:** Select the form name (created in [Creating a New UI Form](#)).
4. Click **Save**. The application instance is created.
5. Publish the application instance to an organization.
 - a. On the Organizations tab of the Application Instance page, click **Assign**.
 - b. In the Select Organizations dialog box, select the organization to which you want to publish the application instance.
 - c. Select the **Apply to entitlements** checkbox.
 - d. Click **OK**.

See Also:

Managing Organizations Associated With Application Instances in *Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed instructions

2.3.1.4 Publishing a Sandbox

You must publish the sandbox that you created in [Creating and Activating a Sandbox](#).

1. Close all the open tabs and pages.

2. In the upper right corner of the page, click the **Sandboxes** link.
The Manage Sandboxes page is displayed.
3. From the table showing the available sandboxes in the Manage Sandboxes page, select the sandbox that you created in [Creating and Activating a Sandbox](#).
4. On the toolbar, click **Publish Sandbox**. A message is displayed asking for confirmation.
5. Click **Yes** to confirm. The sandbox is published and the customizations it contained are merged with the main line.

2.3.2 Changing to the Required Input Locale

Changing to the required input locale (language and country setting) involves installing the required fonts and setting the required input locale.

You may require the assistance of the system administrator to change to the required input locale.

2.3.3 Clearing Content Related to Connector Resource Bundles from the Server Cache

When you deploy the connector, the resource bundles are copied from the resources directory on the installation media into the Oracle Identity Manager database. Whenever you add a new resource bundle to the connectorResources directory or make a change in an existing resource bundle, you must clear content related to connector resource bundles from the server cache.

To clear content related to connector resource bundles from the server cache:

1. In a command window, switch to the *OIM_HOME/server/bin* directory.

 **Note:**

You must perform Step 1 before you perform Step 2. An exception is thrown if you run the command described in Step 2 as follows:

```
OIM_HOME/server/bin/SCRIPT_FILE_NAME
```

2. Enter the following commands:

 **Note:**

You can use the PurgeCache utility to purge the cache for any content category. Run `PurgeCache.bat CATEGORY_NAME` on Microsoft Windows or `PurgeCache.sh CATEGORY_NAME` on UNIX. The `CATEGORY_NAME` argument represents the name of the content category that must be purged.

For example, the following commands purge Metadata entries from the server cache:

```
PurgeCache.bat MetaData
PurgeCache.sh MetaData
```

- **On Microsoft Windows:** `PurgeCache.bat All`
- **On UNIX:** `PurgeCache.sh All`

When prompted, enter the user name and password of an account belonging to the SYSTEM ADMINISTRATORS group. In addition, you are prompted to enter the service URL in the following format:

```
t3://OIM_HOST_NAME:OIM_PORT_NUMBER
```

In this format:

- Replace `OIM_HOST_NAME` with the host name or IP address of the Oracle Identity Manager host computer.
- Replace `OIM_PORT_NUMBER` with the port on which Oracle Identity Manager is listening.

2.3.4 Setting up the Configuration Lookup Definition

Depending on the requirements in your environment, you can add one or more of these additional configuration entries to the `Lookup.FlatFile.Configuration` and `Lookup.FlatFile.Configuration.Trusted` lookup definitions.

Table 2-2 Additional Configuration Entries for the `Lookup.FlatFile.Configuration` and `Lookup.FlatFile.Configuration.Trusted` Lookup Definitions

Code Key	Decode
customConfigParams	Enter the custom configuration parameters required by the custom parser. Sample Value: <code>Type=DOM;Version=1.0</code> For more information about this entry, see Configuring Custom Parsers .
parserClassName	Enter the custom parser implementation class name. If this entry is not present, then by default, the CSV parser implementation is triggered. Sample value: <code>com.extension.parser.XMLParser</code> For more information about this entry, see Configuring Custom Parsers .

Table 2-2 (Cont.) Additional Configuration Entries for the Lookup.FlatFile.Configuration and Lookup.FlatFile.Configuration.Trusted Lookup Definitions

Code Key	Decode
preProcessClassName	Enter the preprocess handler implementation class name. Sample value: <code>com.extension.parser.PreProcessHandler</code> For more information about this entry, see Configuring Preprocess and Postprocess Tasks .
postProcessClassName	Enter the postprocess handler implementation class name. Sample value: <code>com.extension.parser.PostProcessHandler</code> For more information about this entry, see Configuring Preprocess and Postprocess Tasks .
commentCharacter	Enter the character which denotes comment line. Sample value: # For more information about this entry, see Configuring the Connector to Ignore Comment Characters .
progressCheckPoint	Enter the number of successfully processed records after which you require the connector to log the information to denote the progress. Sample Value: 100
encoding	Enter the basic encoding of the file. The default value is the default character set of JVM. Sample value: UTF-8
Process Recon Event Batch Size	Batch size threshold limit for spawning new thread for processing and adding recon event. Default value: 500
Process Recon Event Thread Pool Size	Maximum number of threads that can be created. Default value: 5

2.3.5 Managing Logging

Oracle Identity Manager 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](#)

2.3.5.1 Understanding Log Levels

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

To specify the type of event for which you want logging to take place, you can set the log level to one of the following:

- SEVERE.intValue()+100
This level enables logging of information about fatal errors.
- SEVERE
This level enables logging of information about errors that might allow Oracle Identity Manager to continue running.
- WARNING
This level enables logging of information about potentially harmful situations.
- INFO
This level enables logging of messages that highlight the progress of the application.
- CONFIG
This level enables logging of information about fine-grained events that are useful for debugging.
- FINE, FINER, FINEST
These levels enable logging of information about fine-grained events, where FINEST logs information about all events.

These log levels are mapped to ODL message type and level combinations as shown in [Table 2-3](#).

Table 2-3 Log Levels and ODL Message Type:Level Combinations

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

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

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

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

2.3.5.2 Enabling Logging

To enable logging in Oracle WebLogic Server:

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

```

<log_handler name='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>

```

- b. Replace both occurrences of [LOG_LEVEL] with the ODL message type and level combination that you require. [Table 2-3](#) 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 Manager, all messages generated for this connector that are of a log level equal to or higher than the NOTIFICATION:1 level are recorded in the specified file.

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

For Microsoft Windows:

```
set WLS_REDIRECT_LOG=FILENAME
```

For UNIX:

```
export WLS_REDIRECT_LOG=FILENAME
```


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

4. Restart the application server.

2.3.6 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 to the local computer.
5. Extract the contents of the archive, and open one of the following files in a text editor:
 - For Oracle Identity Manager 11g Release 2 PS2 (11.1.2.2.0):
`SAVED_LOCATION\xliffBundles\oracle\iam\ui\runtime\BizEditorBundle_en.xlf`
 - For releases prior to Oracle Identity Manager 11g Release 2 PS2 (11.1.2.2.0):
`SAVED_LOCATION\xliffBundles\oracle\iam\ui\runtime\BizEditorBundle.xlf`
6. Edit the .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 Manager*, for more information about exporting and importing metadata files

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

2.3.7 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.

To do so:

1. Log in to the Design Console.
2. Expand **Administration**, and then double-click **Lookup Definition**.
3. Depending on whether you have configured your flat file as a target resource or a trusted source, search for and open one of the following lookup definitions:
 - For target resource: **Lookup.FlatFile.Configuration**

- For trusted source: **Lookup.FlatFile.Configuration.Trusted**
4. Click **Add**.
 5. In the newly added row, enter the following values:
 - Code Key: `commentCharacter`
 - Decode: Character that denotes a comment line.
Sample value: `#`
 6. Click **Save**.

2.3.8 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 2-3](#) shows the IT resource values added on the Create IT Resource page.

Figure 2-3 Step 1: Provide IT Resource Information

Create IT Resource

1 2 3 4 5 6

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 Clear

Remote Manager Clear

Cancel Continue >>

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 2-4](#) shows the Step 2: Specify IT Resource Parameter Values page.

Figure 2-4 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

Cancel << Back Continue >>

Figure 2-7 provides information about the parameters of the IT resource.

Table 2-4 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
Timeout	Enter an integer value which specifies the number of milliseconds after which the connection between the connector server and Oracle Identity Manager times out. Sample value: 300
UseSSL	Enter <code>true</code> to specify that you will configure SSL between Oracle Identity Manager and the Connector Server. Otherwise, enter <code>false</code> . Default value: <code>false</code> 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 <code>/setKey [key]</code> option. The value of this key must be specified as the value of the Key IT resource parameter of the connector server.

- 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 2-5](#) shows the Step 3: Set Access Permission to IT Resource page.

Figure 2-5 Step 3: Set Access Permission to IT Resource

Create IT Resource

1 2 3 4 5 6

Step 3 : Set Access Permission to IT Resource

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

Results 1-10 of 19 First | Previous | **Next** | Last

Administrative Role	Display Name	Read Access	Write Access	Delete Access	Unassign
SYSTEM ADMINISTRATORS	SYSTEM ADMINISTRATORS	✓	✓	✓	<input type="checkbox"/>
IDENTITY USER ADMINISTRATORS	IDENTITY USER ADMINISTRATORS	✓	✓	✓	<input type="checkbox"/>
ROLE ADMINISTRATORS	ROLE ADMINISTRATORS	✓	✓	✓	<input type="checkbox"/>
REQUEST ADMINISTRATORS	REQUEST ADMINISTRATORS	✓	✓	✓	<input type="checkbox"/>
RECONCILIATION ADMINISTRATORS	RECONCILIATION ADMINISTRATORS	✓	✓	✓	<input type="checkbox"/>
ATTESTATION EVENT ADMINISTRATORS	ATTESTATION EVENT ADMINISTRATORS	✓	✓	✓	<input type="checkbox"/>
APPROVAL POLICY ADMINISTRATORS	APPROVAL POLICY ADMINISTRATORS	✓	✓	✓	<input type="checkbox"/>
ATTESTATION CONFIGURATION ADMINISTRATORS	ATTESTATION CONFIGURATION ADMINISTRATORS	✓	✓	✓	<input type="checkbox"/>
USER CONFIGURATION ADMINISTRATORS	USER CONFIGURATION ADMINISTRATORS	✓	✓	✓	<input type="checkbox"/>
RESOURCE ADMINISTRATORS	RESOURCE ADMINISTRATORS	✓	✓	✓	<input type="checkbox"/>

Unassign

First | Previous | **Next** | Last

Assign Role Update Permissions

Cancel << Back Continue >>

- 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.
- To proceed with the creation of the IT resource, click **Continue**. Figure 2-6 shows Step 4: Verify IT Resource Details page.

Figure 2-6 Step 4: Verify IT Resource Details

Create IT Resource 1 2 3 4 5 6

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 2-7 shows the Step 5: IT Resource Connection Result page.

Figure 2-7 Step 5: IT Resource Connection Result

Create IT Resource

1 2 3 4 **5** 6

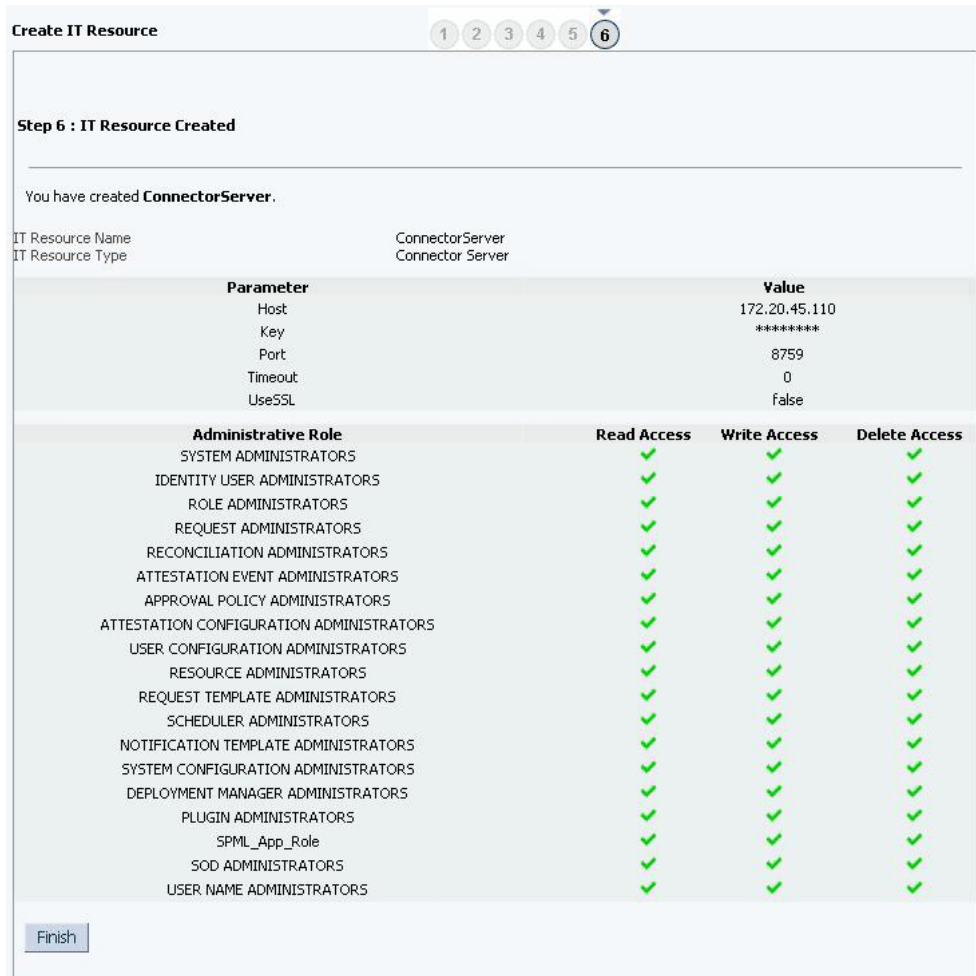
Step 5 : IT Resource Connection Result

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 2-8](#) shows the IT Resource Created page.

Figure 2-8 Step 6: IT Resource Created



2.4 Configuring the Connector with the Target Resource

This section explains how the Flat File connector can be used with a target resource. A target resource can be either a connected resource or a disconnected resource. The Flat File connector will reuse the target connector's Resource Object and Recon Attribute Map for the reconciliation field mappings. This requires the existing connector's application instance name to be given in the scheduled job and pointing the Recon Attribute Map of the flat file connector to the Recon Attribute Map of the target resource.

The following are the prerequisites with respect to configuring the connector for a target resource:

- Ensure that the connector for the target resource that you are using is installed.
- Ensure that the Flat File connector is installed. See [Running the Connector Installer](#) for more information on installing the connector.
- Create a form and an application instance for the target resource that you are using.

- Ensure that the IT Resource for the Flat File connector has been configured. See [Configuring the IT Resource](#) for more information on configuring the IT Resource.

To configure the connector for a target resource:

1. Update the `Lookup.FlatFile.EntFieldMap` lookup definition as follows:
 - a. Log in to the Design Console.
 - b. Expand **Administration** and then double-click **Lookup Definition**.
 - c. Search for and open the **Lookup.FlatFile.EntFieldMap** lookup definition.
 - d. The Code Key column of the target lookup definition is represented by the CODE entry in this lookup definition, which is mapped to NameAttribute (`__NAME__`) mentioned in the schema file. Edit this entry to point to any other field in the flat file.
 - e. The Decode Key column of the target lookup definition is represented by the DECODE entry in this lookup definition, which is mapped to NameAttribute (`__NAME__`) mentioned in the schema file. Edit this entry to point to any other field in the flat file.
 - f. Click **Save**.

2. If you have configured your target resource as trusted source, then:
 - a. Expand **Administration** and then double-click **Lookup Definition**.
 - b. Search for and open the **Lookup.FlatFile.UM.Configuration.Trusted** lookup definition.
 - c. Set the decode value of the Recon Attribute Map code key to the name of the lookup definition of the target resource that holds mappings between resource object fields and enterprise target system attributes.

For example, if you are using Microsoft Active Directory as a target resource, then you must set the value of the decode value of the Recon Attribute Map code key to `Lookup.ActiveDirectory.UM.ReconAttrMap.Trusted`
 - d. Add a new entry to this lookup definition as follows:

Code Key: `Recon Attribute Defaults`

Decode: Name of the lookup definition of the target resource that maps reconciliation fields to their default values. Sample Value:
`Lookup.ActiveDirectory.UM.ReconAttrMap.Trusted.Defaults`
 - e. Click **Save**.

3. If you have configured your target resource as a target resource, then:
 - a. Expand **Administration** and then double-click **Lookup Definition**.
 - b. Search for and open the **Lookup.FlatFile.UM.Configuration** lookup definition.
 - c. Set the decode value of the Recon Attribute Map code key to the name of the lookup definition of the target resource that holds mappings between resource object fields and enterprise target system attributes.

For example, if you are using Microsoft Active Directory as a target resource, then you must set the value of the decode value of the Recon Attribute Map code key to `Lookup.ActiveDirectory.UM.ReconAttrMap`.
 - d. Click **Save**.

2.5 Upgrading the Connector

Upgrading to this release of the connector from earlier releases is not supported.

3

Using the Connector

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

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

- [Performing First-Time Reconciliation](#)
- [Scheduled Jobs for Lookup Field and Entitlement Synchronization](#)
- [Configuring Reconciliation](#)
- [Configuring Scheduled Jobs](#)
- [Reconciling Complex Multivalued Data](#)
- [Configuring Fault Handling](#)
- [Configuring Archival](#)
- [Understanding and Configuring Delimiters](#)
- [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.

3.1 Performing First-Time Reconciliation

First-time reconciliation involves synchronizing lookup definitions in Oracle Identity Manager with the lookup fields of the flat file, and performing full reconciliation. In full reconciliation, all existing user records from the flat file are brought into Oracle Identity Manager.

The following is the sequence of steps involved in reconciling all existing user records:

1. Perform lookup fields and entitlements synchronization by running the scheduled jobs provided for this operation.

See [Scheduled Jobs for Lookup Field and Entitlement Synchronization](#) for information about the attributes of the scheduled jobs for lookup field synchronization.

See [Configuring Scheduled Jobs](#) for information about running scheduled jobs.

2. Perform user reconciliation by running the scheduled jobs for user reconciliation.

See [Attributes of the Scheduled Jobs](#) for information about the attributes of this scheduled task.

See [Configuring Scheduled Jobs](#) for information about running scheduled jobs.



See Also:

[Configuring Scheduled Jobs](#) for information about attributes of the scheduled job

3.2 Scheduled Jobs for Lookup Field and Entitlement Synchronization

The Flat File Entitlements Loader scheduled job is a specialized lookup field synchronization job which reconciles both lookup values and entitlements from a flat file.

In addition to reconciling the lookups from a flat file, this scheduled 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 scheduled job also supports full and incremental reconciliation of lookup values and entitlements.

If you have configured your flat file as a target resource or disconnected resource by using the metadata generation utility, then the following scheduled jobs are created:

- *IT_RES_NAME FIELD_NAME* Loader

This scheduled job is used to load lookup values from the flat file.

For every attribute specified in the `lookupAttributeList` entry of the `FlatFileConfiguration.groovy` file, a corresponding scheduled job for loading or reconciling lookup values from the flat file is created. This is illustrated by the following example:

Suppose the value of the `itResourceDefName` entry is `ACME`. If the value of the `lookupAttributeList` entry is `['Roles', 'Groups']`, then the connector creates the following scheduled jobs:

- ACME Roles Loader
- ACME Groups Loader

These scheduled jobs are used to load lookup values corresponding to roles and groups from the flat file into Oracle Identity Manager.

- *IT_RES FIELD_NAME* Entitlement Loader

This scheduled job adds the entitlements for lookups and synchronizes the catalog with the entitlements automatically.

For every attribute specified in the `entitlementAttributeList` entry of the `FlatFileConfiguration.groovy` file, a corresponding scheduled job for synchronizing the entitlement that is created with the catalog is created. This is illustrated by the following example:

Suppose the value of the `itResourceDefName` entry is `ACME`. If the value of the `entitlementAttributeList` entry is `["Roles.RoleID", "Groups.GroupName"]`, then the connector creates the following scheduled jobs:

- ACME RoleID Entitlement Loader
- ACME GroupName Entitlement Loader

These scheduled jobs are used to synchronize the entitlements RoleID and GroupName with the catalog.

To perform lookup fields and entitlement synchronization, you must specify values for the attributes of this scheduled job. The attributes for the Flat File Entitlements Loader, *IT_RES FIELD_NAME* Loader, and *IT_RES FIELD_NAME* Entitlement Loader scheduled jobs are the same. [Table 3-1](#) describes the attributes of all the three scheduled jobs. [Configuring Scheduled Jobs](#) describes the procedure to configure scheduled jobs.

Table 3-1 Attributes of the Scheduled Jobs for Lookup Fields and Entitlement Synchronization

Attribute	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 OIM administrator must have read and write permissions on this directory.
Flat File directory	Name and complete path to the directory containing flat files to be parsed. Default value: None Note: The OIM administrator must have read and write permissions on this directory.
Flat File IT Resource Name	The name of the Flat File IT Resource.
Is Entitlement	Enter <code>True</code> if the lookup definition is linked to an Entitlement field (for example, Roles). Enter <code>False</code> 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.
Mapping Lookup Name	Name of the lookup definition that holds mapping of enterprise target system attribute names in the flat file and the Code Key and Decode columns of the lookup. See Lookup.FlatFile.EntFieldMap for more information about this lookup definition.
Mode	Enter <code>Full</code> if you want to clear the existing entries in the lookup definition and add new entries. Enter <code>Incremental</code> if you want to append the values to the existing lookup definition. Note: If the value is already present, this entire lookup entry will be replaced.
Target Application Instance Name	Target Application Instance Name to which the data is to be loaded.
Target Lookup Name	Lookup definition name into which the values returned by the connector are loaded. Sample value: <code>Lookup.ACME.Languages</code>

[Figure 3-1](#) shows the Job Details page of the Flat File Entitlements Loader scheduled job.

Figure 3-1 Job Details Page of the Flat File Entitlements Loader Scheduled Job

Job Details : Flat File Entitlements Loader Apply

Job Information

Job Name Flat File Entitlements Loader
 Task Flat File Entitlements Loader
 * Retries Schedule Type
 Periodic
 Cron
 Single
 No pre-defined schedule

Job Status

Current Status Stopped
 Last Run Start
 Last Run End
 Next Scheduled Run

Parameters

Archive directory
 * Flat File directory
 * Flat File IT Resource Name Flat File Entitlements
 Is Entitlement * Mapping Lookup Name
 Mode
 * Target Application Instance Name
 * Target Lookup Name

3.3 Configuring Reconciliation

Reconciliation involves replicating in Oracle Identity Manager the creation of and modifications to user accounts in the flat file.

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

- [Reconciliation Scheduled Jobs](#)
- [Attributes of the Scheduled Jobs](#)
- [Performing Full and Incremental Reconciliation](#)
- [Limited Reconciliation](#)

3.3.1 Reconciliation Scheduled Jobs

When you run the Connector Installer, the scheduled jobs for reconciliation of user records and accounts are created.

Depending on whether you have installed only the ready-to-use Flat File connector that is shipped with the connector installation media, or created the connector using the metadata generation utility, a set of scheduled jobs are created.

If you have installed only the ready-to-use Flat File connector, then the following scheduled jobs are created:

- Flat File User Loader
- Flat File Users Diff Sync Reconciliation
- Flat File Users Delete Reconciliation
- Flat File Users Delete Sync Reconciliation
- Flat File Accounts Loader
- Flat File Accounts Diff Sync Reconciliation
- Flat File Accounts Delete Reconciliation
- Flat File Accounts Delete Sync Reconciliation

If you have created the Flat File connector using the metadata generation utility, then depending on whether you have configured the flat file as a trusted source, target resource, or disconnected resource, the following scheduled jobs are created, in addition to the scheduled jobs listed earlier:

- For trusted source configuration:
 - *IT_RES_NAME* Flat File User Loaders
 - *IT_RES_NAME* Flat File Users Delete Diff Reconciliation
 - *IT_RES_NAME* Flat File Users Delete Reconciliation
 - *IT_RES_NAME* Flat File Users Delete Sync Reconciliation
- For target resource and disconnected resource configuration:
 - *IT_RES_NAME* Flat File Accounts Loader
 - *IT_RES_NAME* Flat File Accounts Delete Diff Reconciliation
 - *IT_RES_NAME* Flat File Accounts Delete Reconciliation
 - *IT_RES_NAME* Flat File Accounts Delete Sync Reconciliation

In these scheduled job names, *IT_RES_NAME* is replaced with the value of the `itResourceDefName` entry in the `FlatFileConfiguration.groovy` file. For example, while configuring the `FlatFileConfiguration.groovy` file, if you have set the value of the `itResourceDefName` entry to `ACME`, then the scheduled jobs are created with names such as `ACME Flat File User Loaders`, `ACME Flat File Users Delete Reconciliation`, `ACME Flat File Account Delete Diff Reconciliation` and so on.

It is recommended that you use the scheduled jobs prefixed with *IT_RES_NAME*, if you have created the connector using the metadata generation utility. While configuring a scheduled job, except for the attribute related to the flat file location, the connector automatically populates values for all other attributes.

The following scheduled jobs are used to retrieve data directly from your enterprise application. These scheduled jobs are created in addition to all the scheduled jobs listed earlier, when you use metadata generation utility:

- For trusted source configuration
 - *IT_RES_NAME* Trusted Incremental User Reconciliation
 - *IT_RES_NAME* Trusted Resource User Delete Reconciliation
 - *IT_RES_NAME* Trusted Resource User Reconciliation
- For target resource configuration
 - *IT_RES_NAME* Target Incremental User Reconciliation
 - *IT_RES_NAME* Target Resource User Delete Reconciliation
 - *IT_RES_NAME* Target Resource User Reconciliation

In order to use these scheduled jobs, you must implement the ICF-based bundle JAR that can handle Search and Sync operations.

3.3.2 Attributes of the Scheduled Jobs

Learn about the attributes of the scheduled jobs for reconciliation of user records and accounts.

- [Scheduled Jobs for Reconciliation of User Records](#)
- [Scheduled Jobs for Reconciliation of Accounts](#)

3.3.2.1 Scheduled Jobs for Reconciliation of User Records

Use the scheduled jobs described in this section if you have configured the enterprise target system as a trusted source.

- [Flat File Users Loader and *IT_RES_NAME* Flat File Users Loader](#)
- [Flat File Users Diff Sync Reconciliation and *IT_RES_NAME* Flat File Users Delete Diff Reconciliation](#)
- [Flat File Users Delete Reconciliation and *IT_RES_NAME* Flat File Users Delete Reconciliation](#)
- [Flat File Users Delete Sync Reconciliation and *IT_RES_NAME* Flat File Users Delete Sync Reconciliation](#)

3.3.2.1.1 Flat File Users Loader and *IT_RES_NAME* Flat File Users Loader

The Flat File Users Loader or *IT_RES_NAME* Flat File Users Loader scheduled job is used for reconciling users from a flat file and creating corresponding users in Oracle Identity Manager.

[Table 3-2](#) lists the attributes of both these schedule jobs.

Table 3-2 Attributes of the Flat File Users Loader and *IT_RES_NAME* Flat File Users Loader Scheduled Jobs

Attribute	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 OIM administrator must have read and write permissions on this directory.
Filter	Expression for filtering records that must be reconciled by the scheduled job. Sample value: <code>startsWith('email','john')</code> Default value: None See Limited Reconciliation for the syntax of this expression.
Flat File directory	Name and complete path to the directory containing flat files to be parsed. Default value: None Note: The OIM administrator must have read and write permissions on this directory.
Flat File IT Resource Name	The name of the IT resource instance that the connector must use to reconcile user data. Default value: For the Flat File Users Loader scheduled job: <code>Flat File Users</code> For <i>IT_RES_NAME</i> Flat File Users Loader scheduled job: <code>IT_RES_NAME</code>

Table 3-2 (Cont.) Attributes of the Flat File Users Loader and IT_RES_NAME Flat File Users Loader Scheduled Jobs

Attribute	Description
Incremental Recon Attribute	<p>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.</p> <p>Sample value: ModifiedDate</p> <p>Default value: None</p>
Latest Token	<p>This attribute holds the value of the column specified as the value of the Incremental Recon Attribute.</p> <p>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 only user accounts that have been modified after the time stamp specified as the value of this attribute are reconciled.</p> <p>If you want to perform a full reconciliation, clear the value in this field.</p> <p>Default value: None</p>
Scheduled Task Name	<p>This attribute holds the name of the scheduled task.</p> <p>Default value:</p> <p>For the Flat File Users Loader scheduled job: Flat File Users Loader</p> <p>For the <i>IT_RES_NAME</i> Flat File Users Loader scheduled job: <i>IT_RES_NAME</i> Flat File Users Loader</p>
Target IT Resource Name	<p>Name of the IT Resource with which the users are associated.</p> <p>Sample value: ACME</p>
Target Resource Object Name	<p>The name of the resource object with which the returned users are associated.</p> <p>Sample value: ACME User Trusted</p>

Figure 3-2 shows the Job Details page of the Flat File Users Loader scheduled job.

Figure 3-2 Job Details Page of the Flat File Users Loader Scheduled Job

The screenshot displays the 'Job Details : Flat File Users Loader' page. It includes an 'Apply' button in the top right corner. The page is divided into several sections:

- Job Information:** Shows Job Name (Flat File Users Loader), Task (Flat File Users Loader), and Retries (0). The Schedule Type is set to 'No pre-defined schedule' (radio button selected).
- Job Status:** Shows Current Status (Stopped), Last Run Start, Last Run End, and Next Scheduled Run.
- Parameters:** Includes fields for Archive directory, Filter, Flat File directory (/home/jm/files/acme/users), Flat File IT Resource Name (Flat File Users), Incremental Recon Attribute, Latest Token, Scheduled Task Name (Flat File Users Loader), Target IT Resource Name (ACME Trusted), and Target Resource Object Name (ACME Trusted User).

3.3.2.1.2 Flat File Users Diff Sync Reconciliation and *IT_RES_NAME* Flat File Users Delete Diff Reconciliation

The Flat File Users Diff Sync Reconciliation or *IT_RES_NAME* Flat File Users Delete Diff Reconciliation scheduled job is used for performing diff-based reconciliation.

 **Note:**

These scheduled jobs only support the reconciliation of deleted users. Users that were added or modified cannot be reconciled using these scheduled jobs.

These scheduled jobs compare the two flat files and return the deleted users alone. It is used to detect deleted users from flat files for enterprise target systems that do not support the export of only the deleted users. 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 users before delete.
- Current Flat File directory
This is the flat file that is exported from the enterprise target system after users have been deleted in the enterprise target system.

While running these scheduled jobs, the connector will detect the users that are missing in the current flat file by comparing them with the users in the previous flat file, and will generate delete reconciliation events only for the missing users.

[Table 3-3](#) lists the attributes of both these schedule jobs.

Table 3-3 Attributes of the Flat File Users Diff Sync Reconciliation and *IT_RES_NAME* Flat File Users Delete Diff Reconciliation Scheduled Jobs

Attribute	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.
Current Flat File directory	The flat file directory which contains the current records from the enterprise target system. Default value: None
Flat File IT Resource Name	The name of the IT resource instance that the connector must use to reconcile user data. Sample value: Flat File Users
Previous Flat File directory	The flat file directory which contains the records from the enterprise target system that were present previously. Default value: None

Table 3-3 (Cont.) Attributes of the Flat File Users Diff Sync Reconciliation and IT_RES_NAME Flat File Users Delete Diff Reconciliation Scheduled Jobs

Attribute	Description
Scheduled Task Name	This attribute holds the name of the scheduled task. Default value: For the Flat File Users Diff Sync Reconciliation scheduled job: Flat File Users Diff Sync Reconciliation For the <i>IT_RES_NAME</i> Flat File Users Delete Diff Reconciliation scheduled job: <i>IT_RES_NAME</i> Flat File Users Delete Diff Reconciliation
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>
Target IT Resource Name	IT Resource with which the users are associated. Sample value: ACME
Target Resource Object Name	The name of the Resource Object with which the returned users are associated. Sample value: ACME User Trusted

3.3.2.1.3 Flat File Users Delete Reconciliation and *IT_RES_NAME* Flat File Users Delete Reconciliation

The Flat File Users Delete Reconciliation or *IT_RES_NAME* Flat File Users Delete Reconciliation scheduled job is used to reconcile data about deleted users in the trusted source (identity management) mode of the connector. During a reconciliation run, for each user account deleted from the enterprise target system, the corresponding OIM User is deleted.

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

 **Note:**

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

Table 3-4 lists the attributes of both these schedule jobs.

Table 3-4 Attributes of the Flat File Users Delete Reconciliation and IT_RES_NAME Flat File Users Delete Reconciliation Scheduled Jobs

Attribute	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 OIM administrator must have read and write permissions on this directory.
Flat File directory	Name and complete path to the directory containing flat files to be parsed. Default value: None Note: The OIM administrator must have read and write permissions on this directory.
Flat File IT Resource Name	The name of the IT resource instance that the connector must use to reconcile user data. Default value: For the Flat File Users Delete Reconciliation scheduled job: Flat File Users For the <i>IT_RES_NAME</i> Flat File Users Delete Reconciliation scheduled job: <i>IT_RES_NAME</i>
Target IT Resource Name	IT Resource with which the users are associated. Sample value: ACME
Target Resource Object Name	The name of the Resource Object with which the returned users are associated. Sample value: ACME User Trusted

3.3.2.1.4 Flat File Users Delete Sync Reconciliation and *IT_RES_NAME* Flat File Users Delete Sync Reconciliation

The Flat File Users Delete Sync Reconciliation or *IT_RES_NAME* Flat File Users Delete Sync Reconciliation scheduled 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 3-5](#) lists the attributes of both these schedule jobs.

Table 3-5 Attributes of the Flat File Users Delete Sync Reconciliation and IT_RES_NAME Flat File Users Delete Sync Reconciliation Scheduled Jobs

Attribute	Description
Archive directory	<p>Enter the name of the directory in which the processed flat files must be saved.</p> <p>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.</p> <p>Note: The OIM administrator must have read and write permissions on this directory.</p>
Delete Attribute	<p>Enter the name of the column in the flat file that represents whether a user is deleted or not.</p> <p>Enter a value for this attribute if you want to perform filtered delete reconciliation.</p> <p>Default value: None</p> <p>Sample value: isDeleted</p>
Delete Attribute Value	<p>Enter the value that is mentioned in the column, which specifies whether a user has been deleted.</p> <p>This column is the value that you specified as the value of the Delete Attribute attribute.</p> <p>Sample value: Yes</p>
Flat File directory	<p>Name and complete path to the directory containing flat files to be parsed.</p> <p>Default value: None</p> <p>Note: The OIM administrator must have read and write permissions on this directory.</p>
Flat File IT Resource Name	<p>The name of the IT resource instance that the connector must use to reconcile user data.</p> <p>Sample value: Flat File Users</p>
Scheduled Task Name	<p>This attribute holds the name of the scheduled task.</p> <p>Default value:</p> <p>For the Flat File Users Delete Sync Reconciliation scheduled job: Flat File Users Delete Sync Reconciliation</p> <p>For the <i>IT_RES_NAME</i> Flat File Users Delete Sync Reconciliation scheduled job: <i>IT_RES_NAME</i> Flat File Users Delete Sync Reconciliation</p>
Sync Token	<p>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.</p> <p>Sample value: <String>123454502019<String></p>
Target IT Resource Name	<p>IT Resource with which the users are associated.</p> <p>Sample value: ACME</p>
Target Resource Object Name	<p>The name of the Resource Object with which the returned users are associated.</p> <p>Sample value: ACME User Trusted</p>

3.3.2.2 Scheduled Jobs for Reconciliation of Accounts

Use the scheduled jobs described in this section if you have configured the enterprise target system as a target resource.

- [Flat File Accounts Loader and *IT_RES_NAME* Flat File Accounts Loader](#)
- [Flat File Accounts Diff Sync Reconciliation and *IT_RES_NAME* Flat File Accounts Delete Diff Reconciliation](#)
- [Flat File Accounts Delete Reconciliation and *IT_RES_NAME* Flat File Accounts Delete Reconciliation](#)
- [Flat File Accounts Delete Sync Reconciliation and *IT_RES_NAME* Flat File Accounts Delete Sync Reconciliation](#)

3.3.2.2.1 Flat File Accounts Loader and *IT_RES_NAME* Flat File Accounts Loader

The Flat File Accounts Loader or *IT_RES_NAME* Flat File Accounts Loader scheduled job is used for reconciling accounts from a flat file and creating corresponding accounts in Oracle Identity Manager.

[Table 3-6](#) lists the attributes of both these schedule jobs.

Table 3-6 Attributes of the Flat File Accounts Loader and *IT_RES_NAME* Flat File Accounts Loader Scheduled Jobs

Attribute	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 OIM administrator must have read and write permissions on this directory.
Filter	Expression for filtering records that must be reconciled by the scheduled job. Sample value: <code>startsWith('email','john')</code> Default value: None See Limited Reconciliation for the syntax of this expression.
Flat File directory	Name and complete path to the directory containing flat files to be parsed. Default value: None Note: The OIM administrator must have read and write permissions on this directory.
Flat File IT Resource Name	The name of the IT resource instance that the connector must use to reconcile user data. Default value: For the Flat File Accounts Loader scheduled job: <code>Flat File Accounts</code> For the <i>IT_RES_NAME</i> Flat File Accounts Loader scheduled job: <code>IT_RES_NAME</code>

Table 3-6 (Cont.) Attributes of the Flat File Accounts Loader and IT_RES_NAME Flat File Accounts Loader Scheduled Jobs

Attribute	Description
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 Default value: None
Latest Token	This attribute 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 only user accounts that have been modified after the time stamp specified as the value of this attribute are reconciled. If you want to perform a full reconciliation, clear the value in this field. Default value: None
Scheduled Task Name	This attribute holds the name of the scheduled task. Default value: For the Flat File Accounts Loader scheduled job: Flat File Accounts Loader For the <i>IT_RES_NAME</i> Flat File Accounts Loader scheduled job: <i>IT_RES_NAME</i> Flat File Accounts Loader
Target Application Instance Name	Target Application Instance Name to which the data is to be loaded. Sample value: ACMEApp

3.3.2.2.2 Flat File Accounts Diff Sync Reconciliation and *IT_RES_NAME* Flat File Accounts Delete Diff Reconciliation

The Flat File Accounts Diff Sync Reconciliation or *IT_RES_NAME* Flat File Accounts Delete Diff Reconciliation scheduled job is used for performing diff-based reconciliation.

 **Note:**

These scheduled jobs only support the reconciliation of deleted accounts. Accounts that were added or modified cannot be reconciled using these scheduled jobs.

These scheduled jobs compare the two flat files and return the deleted accounts alone. They are 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.

While running these scheduled jobs, 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 3-7 lists the attributes of both these schedule jobs.

Table 3-7 Attributes of the Flat File Accounts Diff Sync Reconciliation and IT_RES_NAME Flat File Accounts Delete Diff Reconciliation Scheduled Jobs

Attribute	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.
Filter	Expression for filtering records that must be reconciled by the scheduled job. Sample value: <code>startsWith('email','john')</code> Default value: None See Limited Reconciliation for the syntax of this expression.
Current Flat File directory	The flat file directory which contains the current records from the enterprise target system. Default value: None
Flat File IT Resource Name	The name of the IT resource instance that the connector must use to reconcile user data. Default value: None
Previous Flat File directory	The flat file directory which contains the records from the enterprise target system that were present previously. Default value: None
Scheduled Task Name	This attribute holds the name of the scheduled task. Default value: For the Flat File Accounts Diff Sync Reconciliation scheduled job: Flat File Accounts Diff Sync Reconciliation For the <i>IT_RES_NAME</i> Flat File Accounts Delete Diff Reconciliation scheduled job: <i>IT_RES_NAME</i> Flat File Accounts Delete Diff Reconciliation
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: <code><String>123454502019<String></code>
Target Application Instance Name	Target Application Instance Name to which the data is to be loaded. Sample value: ACMEApp

3.3.2.2.3 Flat File Accounts Delete Reconciliation and *IT_RES_NAME* Flat File Accounts Delete Reconciliation

The Flat File Accounts Delete Reconciliation or *IT_RES_NAME* Flat File Accounts Delete Reconciliation scheduled job is used to reconcile data about deleted accounts in the target resource (account management) mode of the connector. During a reconciliation run, for each account deleted on the enterprise target system, the corresponding OIM account is deleted.

Use these schedule jobs 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 Manager has to verify all the records from the flat file and compare it with existing records to identify if each record has been deleted or not.

Table 3-8 lists the attributes of both these schedule jobs.

Table 3-8 Attributes of the Flat File Accounts Delete Reconciliation and *IT_RES_NAME* Flat File Accounts Delete Reconciliation Scheduled Jobs

Attribute	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 "archive" within the directory containing the flat file, and the processed files are saved in this location. Note: The OIM administrator must have read and write permissions on this directory.
Flat File directory	Name and complete path to the directory containing flat files to be parsed. Default value: None Note: The OIM administrator must have read and write permissions on this directory.
Flat File IT Resource Name	The name of the IT resource instance that the connector must use to reconcile user data. Default value: For the Flat File Accounts Delete Reconciliation scheduled job: Flat File Accounts For the <i>IT_RES_NAME</i> Flat File Accounts Delete Reconciliation scheduled job: <i>IT_RES_NAME</i>
Target Application Instance Name	Target Application Instance Name to which the data is to be loaded. Sample value: ACMEApp

3.3.2.2.4 Flat File Accounts Delete Sync Reconciliation and *IT_RES_NAME* Flat File Accounts Delete Sync Reconciliation

The Flat File Accounts Delete Sync Reconciliation or *IT_RES_NAME* Flat File Accounts Delete Sync Reconciliation scheduled 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 3-9 lists the attributes of both these schedule jobs.

Table 3-9 Attributes of the Flat File Accounts Delete Sync Reconciliation and *IT_RES_NAME* Flat File Accounts Delete Sync Reconciliation Scheduled Jobs

Attribute	Description
Archive directory	<p>Enter the name of the directory in which the processed flat files must be saved.</p> <p>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.</p> <p>Note: The OIM administrator must have read and write permissions on this directory.</p>
Delete Attribute	<p>Enter the name of the column in the flat file that represents whether an account is deleted or not.</p> <p>Enter a value for this attribute if you want to perform filtered delete reconciliation.</p> <p>Default value: None</p> <p>Sample value: isDeleted</p>
Delete Attribute Value	<p>Enter the value that is mentioned in the column, which specifies whether an account has been deleted.</p> <p>This column is the value that you specified as the value of the Delete Attribute attribute.</p> <p>Sample value: Yes</p>
Flat File directory	<p>Name and complete path to the directory containing flat files to be parsed.</p> <p>Default value: None</p> <p>Note: The OIM administrator must have read and write permissions on this directory.</p>
Flat File IT Resource Name	<p>The name of the IT resource instance that the connector must use to reconcile user data.</p> <p>Default value:</p> <p>For the Flat File Accounts Delete Sync Reconciliation scheduled job: Flat File Accounts</p> <p>For the <i>IT_RES_NAME</i> Flat File Accounts Delete Sync Reconciliation: <i>IT_RES_NAME</i></p>

Table 3-9 (Cont.) Attributes of the Flat File Accounts Delete Sync Reconciliation and IT_RES_NAME Flat File Accounts Delete Sync Reconciliation Scheduled Jobs

Attribute	Description
Scheduled Task Name	This attribute holds the name of the scheduled task. Default value: For the Flat File Accounts Delete Sync Reconciliation scheduled job: Flat File Accounts Delete Sync Reconciliation For the <i>IT_RES_NAME</i> Flat File Accounts Delete Sync Reconciliation scheduled job: <i>IT_RES_NAME</i> Flat File Accounts Delete Sync Reconciliation
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>
Target Application Instance Name	Target Application Instance Name to which the data is to be loaded. Sample value: ACMEApp

3.3.3 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 may use a diff-based approach to reconcile the deleted records into Oracle Identity Manager.
- For entitlements reconciliation, the scheduled job for loading entitlements can be run in an incremental or full mode. See [Scheduled Jobs for Lookup Field and Entitlement Synchronization](#) for more information about the scheduled jobs available for entitlement loading and the attributes.

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

After you deploy the connector, 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 Manager.

To perform a full reconciliation run, ensure that no values are specified for the Latest Token, Incremental Recon Attribute, and Filter attributes of the scheduled jobs for Users Loader or Accounts Loader for reconciling user records or accounts respectively.

At the end of the reconciliation run, the Latest Token attribute of the scheduled job for user record or account reconciliation is automatically set to the most recent value obtained from the attribute (for incremental recon) of the flat file. This happens only if you have configured the connector for incremental reconciliation as discussed in

[Configuring the Connector for Incremental Reconciliation](#). From the next reconciliation run onward, only records created or modified after this most recent value are considered for reconciliation. If you have not configured your connector for incremental reconciliation, then the Latest Token attribute remains blank and the connector continues to perform full reconciliation runs.

3.3.4 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.

You do this by creating filters for the reconciliation module. The connector supports filters in the reconciliation scheduled 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 the reconciliation module. This connector provides a Filter attribute (a scheduled task attribute) that allows you to use any of the Flat File resource attributes to filter the target system records.

See [About Filters](#) for information about implementing filters in the custom parser.

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



Note:

The `__UID__` attribute name can only be used with the `equalTo` filter.

While deploying the connector, follow the instructions in [Configuring Scheduled Jobs](#) to specify attribute values.

3.4 Configuring Scheduled Jobs

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

You can apply this procedure to configure the scheduled jobs for lookup field synchronization and reconciliation.

To configure a scheduled job:

1. Log in to Oracle 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 following parameters:
 - **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.

 **Note:**

See *Creating Jobs in Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed information about schedule types.

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:**

- Attribute values are predefined in the connector XML file that you import. Specify values only for those attributes that you want to change.
- 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:**

The Stop Execution option is available in the Administrative and User Console. You can use the Scheduler Status page to start, stop, or reinitialize the scheduler.

3.5 Reconciling 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.

The child form data must be in the same 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 are separated by customizable delimiters.

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

```
AccountID,FirstName,LastName,Email,Languages,Roles
```

```
"111", "John", "Doe", "john.doe@acme.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*.

3.6 Configuring Fault Handling

Record level errors while parsing the file are logged in a separate file and will be saved in a directory named "failed" that the connector creates, within the flat file directory.

The processed flat file will be saved 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 will contain all those records that were not processed due to validation or data errors. The connector will also append 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, the same file can be modified to fix the data errors and loaded back using the connector to reconcile the failed records alone. The Oracle Identity Manager Administrator must have read and write permissions on the Flat File directory and Archive directory locations.

3.7 Configuring Archival

The connector supports archival of the processed flat files.

The archive directory location can be specified in the Archive directory attribute while configuring the scheduled jobs, and the connector will move the files from the source directory to the specified location, once each file is processed. If the value for this attribute is not specified, the connector creates a directory named "archived" within the directory containing the flat file, and the processed files are saved in this location. The Oracle Identity Manager Administrator must have read and write permissions on Archive directory location. The processed flat file will be saved in the following format:

FILENAME_dd-MM-yyyy_HH-mm-ss.zip

In this format:

- *FILENAME*
This 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*
This 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, the filename will be saved in the following format:

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

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

See the following sections for more information about the Archive directory attribute:

- [Scheduled Jobs for Lookup Field and Entitlement Synchronization](#)
- [Attributes of the Scheduled Jobs](#)

3.8 Understanding and Configuring Delimiters

Learn about the delimiters that the connector supports and its configuration.

- [About Delimiters Supported By the Connector](#)
- [Configuring Delimiters](#)

3.8.1 About Delimiters Supported By the Connector

The connector supports the use of single character delimiters that can be used to separate values in a record.

The Space or tab characters have to be entered as `space` or `tab` respectively. Other multibyte characters (characters in different locale) can be directly entered in the lookup definition in the respective locale.

Note:

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

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, they must be specified in the `Lookup.FlatFile.Configuration` lookup definition if you have configured your flat file as a target resource, and in the `Lookup.FlatFile.Configuration.Trusted` if you have configured your flat file as a trusted source.

See the `fieldDelimiter`, `multiValueDelimiter`, and `subFieldDelimiter` entries in [Table 1-2](#) and [Table 1-3](#) for more information about delimiters.

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

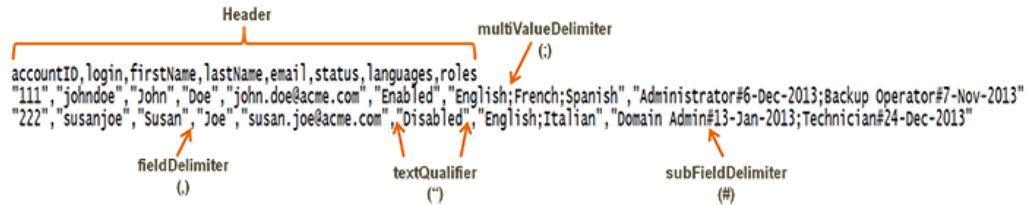
```
AccountID,FirstName,LastName,Email,Languages,Roles
```

```
"111", "John", "Doe", "john.doe@acme.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 3-3 shows sample multivalued data separated by delimiters.

Figure 3-3 Sample Multivalued Data Separated by Delimiters



3.8.2 Configuring Delimiters

Perform this procedure to configure delimiters.

1. Log in to the Design Console.
2. Expand **Administration**, and then double-click **Lookup Definition**.
3. Depending on how you have configured your flat file, perform one of the following steps:
 - If you have configured your flat file as a target resource:
Search for and open the **Lookup.FlatFile.Configuration** lookup definition.
 - If you have configured your flat file as a trusted source:
Search for and open the **Lookup.FlatFile.Configuration.Trusted** lookup definition
4. Depending on the entries that you want to configure, perform the following steps:
 - To configure the fieldDelimiter entry:
If your flat file uses the slash symbol (/) as a field delimiter, specify it in the Decode column, as the value of this entry.
Default value: ,
Sample value: /
 - To configure the multiValueDelimiter entry:
If your flat file uses the vertical bar (|) as a delimiter for each multivalued data, specify it in the Decode column, as the value of this entry.
Default value: ;
Sample value: |
 - To configure the subFieldDelimiter entry:
If your flat file uses the dollar sign (\$) as a delimiter for each subfield within a multivalued field, specify it in the Decode column, as the value of this entry.
Default value: #
Sample value: \$

5. Click **Save**.

3.9 Uninstalling the Connector

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

If you want to uninstall the connector for any reason, see Uninstalling Connectors in *Oracle Fusion Middleware Administering Oracle Identity Manager*.

4

Extending the Functionality of the Connector

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

This chapter contains the following optional procedures:

Note:

From Oracle Identity Manager Release 11.1.2 onward, lookup queries are not supported. See *Managing Lookups in Oracle Fusion Middleware Administering Oracle Identity Manager* for information about managing lookups by using the Form Designer in the Oracle Identity Manager System Administration console.

- [Configuring Custom Parsers](#)
- [Configuring Preprocess and Postprocess Tasks](#)
- [Adding New Attributes for Target Resource Reconciliation](#)
- [Adding New Multivalued Attributes for Target Resource Reconciliation](#)
- [Configuring Validation of Data During Reconciliation](#)
- [Configuring Transformation of Data During Reconciliation](#)
- [Configuring the Connector for Multiple Installations of the Target System](#)

4.1 Configuring 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.

This section contains the following topics:

- [Creating the Custom Parser](#)
- [Integrating the Custom Parser with the Flat File Connector](#)

4.1.1 Creating the Custom Parser

Learn about creating a custom parser.

- [About Creating the Custom Parser](#)
- [About Mandatory Attributes Validation](#)

- [About Filters](#)
- [About Handling Multivalued Attributes](#)

4.1.1.1 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) {

        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 About 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 About 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 About 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
}

```

4.1.1.2 About 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.

4.1.1.3 About 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
}
```

4.1.1.4 About 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>.

4.1.2 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-1.0.1115.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.

 **See Also:**

Downloading JAR Utility in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

3. Update the `org.identityconnectors.flatfile-1.0.1115.jar` file as follows:
 - a. Extract the contents of the `org.identityconnectors.flatfile-1.0.1115.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-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-1.0.1115.jar` file to the database. This utility is copied into the following location when you install Oracle Identity Manager:

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 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 (ICFBundle) as the value of the JAR type.

 **See Also:**

Upload JARs Utility in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*

5. Restart the Oracle Identity Manager server after updating the JAR file.
6. Update the Configuration lookup definition as follows:
 - a. Log in to the Design Console.
 - b. Depending on whether you have configured your target system as a target resource or a trusted source, search for and open one of the following lookup definitions:

For target resource: **Lookup.FlatFile.Configuration**
For trusted source: **Lookup.FlatFile.Configuration.Trusted**
 - c. Click **Add**.
 - d. In the newly added row, enter the following values:

Code Key: `parserClassName`

Decode: The fully qualified name of the class implementing the custom parser.

Sample value: `com.extension.parser.XMLParser`
 - e. If the custom parser uses custom configuration parameters, then click **Add** to add the following entry:

Code Key: `customConfigParams`

Decode: Specify the custom configuration parameters used by the custom parser. The decode value must be in the name-value format. In other words, in the following format:

`NAME1=VALUE1;NAME2=VALUE2`

Sample value: `Type=DOM;Version=1.0`
 - f. Click **Save**.

4.2 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](#)

4.2.1 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](#)

4.2.1.1 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.  
  
    }  
  
}
```

4.2.1.2 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.  
  
}  
  
}
```

4.2.2 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-1.0.1115.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.

See Also:

Downloading JAR Utility in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

3. Update the `org.identityconnectors.flatfile-1.0.1115.jar` file as follows:
 - a. Extract the contents of the `org.identityconnectors.flatfile-1.0.1115.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-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-1.0.1115.jar file to the database. This utility is copied into the following location when you install Oracle Identity Manager:

- **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 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 (ICFBundle) as the value of the JAR type.

 **See Also:**

Upload JARs Utility in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*

5. Restart the Oracle Identity Manager Server after updating the jar.
6. Repeat Steps 1 through 5 to create a JAR file to hold the postprocess task class and copy the JAR file into Oracle Identity Manager.
7. Update the Configuration lookup definition as follows:

- a. Log in to the Design Console.

- b. Depending on whether you have configured your target system as a target resource or a trusted source, search for and open one of the following lookup definitions:

For target resource: **Lookup.FlatFile.Configuration**

For trusted source: **Lookup.FlatFile.Configuration.Trusted**

- c. Click **Add**.

- d. If you have written code for the preprocess task, then in the newly added row, enter the following values:

Code Key: preProcessClassName

Decode: The fully qualified name of the class implementing the preprocess task.

Sample value: `com.extension.parser.PreProcessHandler`

- e. If you have written code for the postprocess task, then in the newly added row, enter the following values:

Code Key: `postProcessClassName`

Decode: The fully qualified name of the class implementing the preprocess task.

Sample value: `com.extension.parser.PostProcessHandler`

- f. Click **Save**.

4.3 Adding New Attributes for Target Resource Reconciliation

By default, the attributes listed in the Recon Attribute Map of the connector are mapped for reconciliation between Oracle Identity Manager and the target system. If required, you can add new attributes for target resource reconciliation.



Note:

This section describes an optional procedure. You need not perform this procedure if you do not want to add new attributes for reconciliation.

To add a new attribute for target resource reconciliation, perform the following procedure:



Note:

You must ensure that the new attributes that you add for reconciliation contain data in string-format only. Binary attributes must not be introduced into Oracle Identity Manager natively.

1. Log in to the Oracle Identity Manager Design Console.
2. Add the new attribute on the OIM User process form as follows:
 - a. Expand **Development Tools**.
 - b. Double-click **Form Designer**.
 - c. Search for and open the process form of the enterprise target system. For example, **UD_ACME_USR**.
 - d. Click **Create New Version**.
 - e. In the **Label** field, enter the version name. For example, `version#1`.
 - f. Click **Save**.

- g. Select the current version created in Step 2.e from the **Current Version** list.
- h. Click **Add** to create a new attribute, and provide the values for that attribute.
For example, if you are adding the address number attribute, then enter the following values in the **Additional Columns** tab:

Field	Value
Name	AddressNumber
Variant Type	String
Length	100
Field Label	AddressNumber
Order	14

- i. Click **Save**.
 - j. Click **Make Version Active**.
3. All changes made to the Form Designer of the Design Console must be done in a new UI form as follows:
 - a. Log in to Oracle Identity System Administration.
 - b. Create and active a sandbox. See [Creating and Activating a Sandbox](#) for more information.
 - c. Create a new UI form to view the newly added field along with the rest of the fields. See [Creating a New UI Form](#) for more information about creating a UI form.
 - d. Associate the newly created UI form with the application instance of your target system. To do so, open the existing application instance for your resource, from the Form field, select the form (created in Step 3.c), and then save the application instance.
 - e. Publish the sandbox. See [Publishing a Sandbox](#) for more information.
 4. Add the new attribute to the list of reconciliation fields in the resource object as follows:
 - a. Expand **Resource Management**.
 - b. Double-click **Resource Objects**.
 - c. Search for and open the target resource object. For example, **ACME User**.
 - d. On the Object Reconciliation tab, click **Add Field**, and then enter the following values:
 - Field Name:** AddressNumber
 - Field Type:** String
 - e. Click **Save** and then close the dialog box.
 5. Create a reconciliation field mapping for the new attribute in the process definition form as follows:
 - a. Expand **Process Management**.
 - b. Double-click **Process Definition**.
 - c. Search for and open the **ACME User** process definition.

- d. On the Reconciliation Field Mappings tab, click **Add Field Map**, and then select the following values:
 - Field Name:** AddressNumber
 - Field Type:** String
 - Process Data Field:** AddressNumber
 - e. Click **Save**.
 - f. Click **Create Reconciliation Profile** from the corresponding resource object. This copies changes made to the resource object into the MDS.
6. Create an entry for the attribute in the lookup definition for reconciliation as follows:
- a. Expand **Administration**.
 - b. Double-click **Lookup Definition**.
 - c. Search for and open the **Recon Attribute Map** lookup definition of the Flat File connector.
 - d. Click **Add** and enter the **Code Key** and **Decode** values for the attribute. The Code Key value must be the name of the attribute given in the resource object. The Decode value is the name of the corresponding column in the flat file.

For example, enter `AddressNumber` in the **Code Key** field and then enter `ADDRS_NUM` in the **Decode** field. Note that both Code Key and Decode values are the same.
 - e. Click **Save**.

4.4 Adding New Multivalued Attributes for Target Resource Reconciliation

You can add new multivalued attributes such as Roles, which can have subfields like role name and so on.

To add a new multivalued attribute for target resource reconciliation:

1. Log in to the Oracle Identity Manager Design Console.
2. Create a form for the multivalued field as follows:
 - a. Expand **Development Tools** and double-click **Form Designer**.
 - b. Create a form by specifying a table name, for example `UD_CARLICEN` and description, and then click **Save**.
 - c. Click **Add** and enter the details of the field.
 - d. Click **Save** and then click **Make Version Active**.
3. Add the form created for the multivalued field as a child form of the process form as follows:
 - a. Search for and open the parent process form. For example, **US_ACMEUSR**.
 - b. Click **Create New Version**.
 - c. Click the **Child Table(s)** tab.
 - d. Click **Assign**.

- e. In the Assign Child Tables dialog box, select the newly created child form, click the right arrow, and then click **OK**.
 - f. Click **Save** and then click **Make Version Active**.
4. All changes made to the Form Designer of the Design Console must be done in a new UI form as follows:
 - a. Log in to Oracle Identity System Administration.
 - b. Create and active a sandbox. See [Creating and Activating a Sandbox](#) for more information.
 - c. Create a new UI form to view the newly added field along with the rest of the fields. See [Creating a New UI Form](#) for more information about creating a UI form.
 - d. Associate the newly created UI form with the application instance of your target system. To do so, open the existing application instance for your resource, from the Form field, select the form (created in Step 4.c), and then save the application instance.
 - e. Publish the sandbox. See [Publishing a Sandbox](#) for more information.
5. Add the new multivalued field to the list of reconciliation fields in the resource object as follows:
 - a. Log in to the Design Console.
 - b. Expand **Resource Management** and then double-click **Resource Objects**.
 - c. Search for and open one the resource objects. For example, **ACME User**.
 - d. On the Object Reconciliation tab, click **Add Field**.
 - e. In the Add Reconciliation Fields dialog box, enter the details of the field.
For example, enter `carlicenses` in the **Field Name** field and select **Multi-Valued Attribute** from the Field Type list.
 - f. Click **Save** and then close the dialog box.
 - g. Right-click the newly created field and select **Define Property Fields**.
 - h. In the Add Reconciliation Fields dialog box, enter the details of the newly created field.
For example, enter `carlicense` in the Field Name field and select **String** from the Field Type list.
 - i. Click **Save** and then close the dialog box.
6. Create an entry for the field in the lookup definition for reconciliation as follows:
 - a. Expand **Administration** and then double-click **Lookup Definition**.
 - b. Search for and open the Recon Attribute Map lookup definition. For example, **Lookup.ACME.UM.ReconAttrMap**.

 **Note:**

For the target system fields, you must use the same case (uppercase or lowercase) as given on the target system. This is because the field names are case-sensitive.

- c. Click **Add** and enter the Code Key and Decode values for the field, and then Click **Save**. The Code Key and Decode values must be in the following format:

Code Key:
`MULTIVALUED_FIELD_NAME~CHILD_RESOURCE_OBJECT_FIELD_NAME`

Decode: Corresponding target system attribute.

For example, enter `carlicenses~carlicense` in the Code Key field and then enter `carlicense` in the Decode field.
7. Create a reconciliation field mapping for the new field as follows:
 - a. Expand **Process Management** and double-click **Process Definition**.
 - b. Search for and open one of the **ACME User** process definitions.
 - c. On the Reconciliation Field Mappings tab of the ACME User process definition, click **Add Table Map**.
 - d. In the Add Reconciliation Table Mapping dialog box, select the field name and table name from the list, click **Save**, and then close the dialog box.
 - e. Right-click the newly created field, and select **Define Property Field Map**.
 - f. In the Field Name field, select the value for the field that you want to add.
 - g. Double-click the **Process Data Field** field, and then select **UD_CARLICEN**.
 - h. Select **Key Field for Reconciliation Field Matching** and click **Save**.
 - i. Click **Create Reconciliation Profile** from the corresponding resource object. This copies changes made to the resource object into the MDS.

4.5 Configuring Validation of Data During Reconciliation

You can configure validation of reconciled 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 (#).

To configure validation of data:

1. Write code that implements the required validation logic in a Java class.

The following sample validation class checks if the value in the First Name attribute contains the number sign (#):

```
package com.validate;
import java.util.*;
public class MyValidation {

public boolean validate(HashMap hmUserDetails,
    HashMap hmEntitlementDetails, String field) {
    /*
    * You must write code to validate attributes. Parent
    * data values can be fetched by using hmUserDetails.get(field)
    * For child data values, loop through the
    * ArrayList/Vector fetched by hmEntitlementDetails.get("Child
    Table")
    * Depending on the outcome of the validation operation,
    * the code must return true or false.
    */
    /*
```



```

* In this sample code, the value "false" is returned if the field
* contains the number sign (#). Otherwise, the value "true" is
* returned.
*/
    boolean valid=true;
    String sFirstName=(String) hmUserDetails.get(field);
    for(int i=0;i<sFirstName.length();i++){
        if (sFirstName.charAt(i) == '#'){
            valid=false;
            break;
        }
    }
    return valid;
}
} /* End */

```

2. Create a JAR file to hold the Java class.
3. Run the Oracle Identity Manager Upload JARs utility to post the JAR file created in Step 2 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

`OIM_HOME/server/bin/UploadJars.bat`

For UNIX:

`OIM_HOME/server/bin/UploadJars.sh`

When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Manager administrator, URL of the Oracle Identity Manager host computer, context factory value, type of JAR file being uploaded, and the location from which the JAR file is to be uploaded. Specify 1 as the value of the JAR type.

4. If you created the Java class for validating a process form field for reconciliation, then:
 - a. Log in to the Design Console.
 - b. Search for and open the **Lookup.FlatFile.UM.ReconValidation** lookup definition.

 **Note:**

If you do not find this lookup definition, then create it.

- c. In the Code Key column, enter the resource object field name that you want to validate For example, `Username`. In the Decode column, enter the class name. For example, `org.identityconnectors.flatfile.extension.ACMEValidator`.

- d. Save the changes to the lookup definition.
 - e. Search for and open the **Lookup.FlatFile.UM.Configuration** lookup definition.
 - f. In the Code Key column, enter Recon Validation Lookup. In the Decode column, enter Lookup.FlatFile.UM.ReconValidation.
 - g. Save the changes to the lookup definition.
5. Purge the cache to get the changes reflected in Oracle Identity Manager. See Purging Cache in *Oracle Fusion Middleware Administering Oracle Identity Manager*.

4.6 Configuring Transformation of Data During Reconciliation

You can configure the transformation of reconciled single-valued 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 Manager.

Note:

- This section describes an optional procedure. Perform this procedure only if you want to configure transformation of data during reconciliation.
- Do not copy the the transformation jar into the connector bundle. Instead, register it as JavaTask only.

To configure the transformation of data:

1. Write code that implements the required transformation logic in a Java class.

This transformation class must implement the transform method. The following sample transformation class modifies the Username attribute by using values fetched from the `__NAME__` attribute of the target system:

```
package com.transformationexample;
import java.util.HashMap;
public class MyTransformer {
public Object transform(HashMap hmUserDetails, HashMap
hmEntitlementDetails, String sField) throws ConnectorException {
    /*
    * You must write code to transform the attributes.
    * Parent data attribute values can be fetched by using
    hmUserDetails.get("Field Name").
    * To fetch child data values, loop through the
    * ArrayList/Vector fetched by hmEntitlementDetails.get("Child Table")
    * Return the transformed attribute.
    */
    String sUserName = (String) hmUserDetails.get("__NAME__");
    return sUserName + "@example.com";
}
}
```

2. Create a JAR file to hold the Java class.

3. Run the Oracle Identity Manager Upload JARs utility to post the JAR file created in Step 2 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

`OIM_HOME/server/bin/UploadJars.bat`

For UNIX:

`OIM_HOME/server/bin/UploadJars.sh`

When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Manager administrator, URL of the Oracle Identity Manager host computer, context factory value, type of JAR file being uploaded, and the location from which the JAR file is to be uploaded. Specify 1 as the value of the JAR type.

4. Create a new lookup definition by the name `Lookup.FlatFile.UM.ReconTransformation` as follows:
 - a. Log in to the Design Console.
 - b. Expand **Administration**, and then double-click **Lookup Definition**.
 - c. In the Code field, enter `Lookup.FlatFile.UM.ReconTransformation` as the name of the lookup definition.
 - d. In the Field field, enter the name of the table column of the Oracle Identity Manager or user-created form or tab, from which the text field, lookup field, or box field will be accessible.
 - e. Select the **Lookup Type** option.
 - f. On the Lookup Code Information tab, click **Add**.
 - g. In the **Code Key** column, enter the name of the attribute on which you want to apply the transformation. For example: `FirstName`.
 - h. In the **Decode** column, enter the name of the class file. For example: `oracle.iam.connectors.flatfile.Transformation`.
 - i. Save the lookup definition.
5. Update the User Configuration lookup definition as follows:
 - a. Expand **Administration**, and then double-click **Lookup Definition**.
 - b. Search for and open the **Lookup.FlatFile.UM.Configuration** lookup definition.
 - c. On the Lookup Code Information tab, click **Add**.
A new row is added.
 - d. In the **Code Key** column, enter `Recon Transformation Lookup`.
In the **Decode** column, enter `Lookup.FlatFile.UM.ReconTransformation`.

- e. Save the changes to the lookup definition.
6. Purge the cache to get the changes reflected in Oracle Identity Manager. See *Purging Cache in Oracle Fusion Middleware Administering Oracle Identity Manager* for information on purging cache.

4.7 Configuring the Connector for Multiple Installations of the Target System

You must create copies of connector objects to configure the connector for multiple installations of the target system.

The following example illustrates this requirement to configure the connector for multiple installations of the target system:

The London and New York offices of Example Multinational deal with flat files of different formats, and want to associate custom parsers, preprocess and postprocess handlers with each of these different formats. The company has recently installed Oracle Identity Manager, and they want to configure Oracle Identity Manager to link all the installations of the target system.

To meet the requirement posed by such a scenario, you can create copies of connector objects, such as the IT resource and resource object.

The decision to create a copy of a connector object is based on a requirement. For example, an IT resource can hold connection information for one target system installation. Therefore, it is mandatory to create a copy of the IT resource for each target system installation.

With some other connector objects, you do not need to create copies at all. For example, a single attribute-mapping lookup definition can be used for all installations of the target system.

To configure the connector for multiple installations of the target system:

1. Create a Flat File connector bundle with a different version. To do so:
 - a. Extract the contents of the `org.identityconnectors.flatfile-1.0.1115.jar` file on the installation media to a temporary directory.
 - b. In a text editor, open the `MANIFEST.MF` file located in the `META-INF` directory for editing.
 - c. Specify a new value for the `ConnectorBundle-Version` attribute. For example, specify `1.0.1116` as the new value.
 - d. Save and close the file.
 - e. Upload the JAR file by performing the procedure described in Step 3 of [Integrating the Custom Parser with the Flat File Connector](#) to specify a custom parser, or perform the procedure described in Step 3 of [Integrating the Preprocess and Postprocess Tasks with the Flat File Connector](#) to specify custom preprocess or postprocess tasks.
 - f. Rename the connector bundle to reflect the new version. For example, `org.identityconnectors.flatfile-1.0.1116.jar`.
2. Run the Oracle Identity Manager Upload JARs utility to upload the newly created JAR file (for example, `org.identityconnectors.flatfile-1.0.1116.jar` file) to 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/UploadJars.bat`

For UNIX:

`OIM_HOME/server/bin/UploadJars.sh`

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

 **See Also:**

Upload JARs Utility in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*

3. Create a configuration lookup definition for this instance of the target system. For example, create a lookup definition by the name `Lookup.FlatFile.Configuration1`.
4. Add the following entries to this lookup definition and specify the corresponding values in the Decode column:
 - Connector Name
 - Bundle Version
 - User Configuration Lookup
 - Bundle Name

 **Note:**

Ensure that the Decode value of Bundle Version is the latest version specified in Step 2. For example, 1.0.1116. For all entries other than Bundle Version, you can specify the same values as those present in the `Lookup.FlatFile.Configuration` lookup definition.

5. Create an IT resource of the Flat File IT Resource type. Ensure that the value of the Configuration Lookup parameter in this newly created IT resource contains the name of the lookup definition created in Step 3.

6. If you are using the connector server, then repeat Steps 1 through 5 of this section with the following difference:

While performing Step 2 of this procedure, instead of uploading the newly created JAR file to Oracle Identity Manager database, copy it to the `CONNECTOR_SERVER_DIR/bundles` directory.

5

Troubleshooting the Connector

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

Table 5-1 Troubleshooting for the Flat File Connector

Problem	Reason	Solution
<p>The following error message is encountered:</p> <pre>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.</pre>	<p>The reason for the error could be any one of the following:</p> <ol style="list-style-type: none">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.	<p>The following are the solutions to their corresponding causes mentioned in the Reason column of this row:</p> <ol style="list-style-type: none">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.
<p>The following error message is encountered:</p> <pre>org.identityconnectors.framework.common.exceptions.ConfigurationException: Directory does not contain normal files to read xxxx</pre>	<p>The flat file directory is empty.</p>	<p>Ensure that the directory is valid and contains files to be parsed.</p>

Table 5-1 (Cont.) Troubleshooting for the Flat File Connector

Problem	Reason	Solution
<p>The following error message is encountered:</p> <pre>org.identityconnectors.framework.common.exceptions.ConnectorException: Specified class not found xxxxxx</pre>	<p>The reason for the error could be any one of the following:</p> <ol style="list-style-type: none"> <li data-bbox="695 443 967 726">1. The value given for the parserClassName, preProcessClassName, or postProcessClassName entries of the main configuration lookup definition is invalid. <li data-bbox="695 747 967 831">2. The custom class is not available to the class loader. 	<p>The following are the solutions to their corresponding causes mentioned in the Reason column of this row:</p> <ol style="list-style-type: none"> <li data-bbox="984 474 1378 663">1. Ensure that the fully qualified class name is mentioned for the parserClassName, preProcessClassName, or postProcessClassName entries of the main configuration lookup definition. <li data-bbox="984 684 1378 800">2. Upload the custom parser, preprocess, or postprocess handler JARs to Oracle Identity Manager.

6

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.

7

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](#)
- [Error Encountered When Process Recon Event Batch Size is Set to a Smaller Value](#)

7.1 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.

7.2 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**.

7.3 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.

7.4 Error Encountered When Process Recon Event Batch Size is Set to a Smaller Value

This error is encountered when the multithreading configuration entry "Process Recon Event Batch Size" is set to smaller values.

```
oracle.iam.platform.tx.OIMTransactionException:  
java.sql.SQLIntegrityConstraintViolationException: ORA-00001: unique  
constraint (XXXXXX.PK_RECON_JOBS) violated
```

This is an issue associated with Oracle Identity Manager.

Workaround: Reduce the value of the Process Recon Event Thread Pool Size entry (default value) and increase the value of the Process Recon Event Batch Size entry.

Process Recon Event Thread Pool Size and Process Recon Event Batch Size are the entries in the main configuration lookup definitions such as Lookup.FlatFile.Configuration and Lookup.FlatFile.Configuration.Trusted. See [Setting up the Configuration Lookup Definition](#) for more information about these entries.

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","12
20227200000","1420215120000"
```

```
"SDOE","Susan.Doe","Susan","Doe","susan.joe@example.com","INR","54678","Disabled"
,"1220227200000","1420128720000"
```

```
"MJOE","Mikeal.Joe","Mikeal","Joe","mikeal.joe@example.com","EUR","43242","Enable
d","1220227200000","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,JoiningDa
te,LastUpdated
```

```
"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","14202151200
00"
```

```
"SDOE","Susan.Doe","Susan","Doe","susan.joe@example.com","INR","54678","Sports
Group","Advanced User#1220227200000#1220227200000;Group
Administrator#1220227200000#1220227200000","Disabled","1220227200000","1420128720
```

000"

"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

Sample Schema Files

These are the sample schema files.

- [Sample Schema File for Users](#)
- [Sample Schema File for Currency](#)
- [Sample Schema File for Groups](#)
- [Sample Schema File for Roles](#)

B.1 Sample Schema File for Users

The section displays is a snippet of a schema file for reconciling users.

This schema file supports incremental reconciliation by using the "LastUpdated" attribute. As discussed earlier in [Configuring the Connector for Incremental Reconciliation](#), the attribute being used for incremental reconciliation (for example, LastUpdated) has been listed in the FieldNames qualifier and its datatype has been set to Long.

```
#Schema file for ACME User

#List of fields
FieldNames=UID,UserId,FirstName,LastName,email,Currency,Salary,status,JoiningDate
,LastUpdated,Groups,Roles

#Unique ID Attribute
UidAttribute=UID

#Account Name attribute
NameAttribute=UserId

#Multivalued attributes
Groups.Multivalued=true
Roles.Multivalued=true

#Subfields for complex child form
Roles.Subfields=RoleName,Start_Date,End_Date

#Complex child form objectClass
Roles.EmbeddedObjectClass=MyROLES

#Datatypes (Default:String)
Roles.Start_Date.DataType=Long
Roles.End_Date.DataType=Long
FirstName.DataType=String
JoiningDate.DataType=Long

#Incremental reconciliation attribute with datatype set to Long
LastUpdated.DataType=Long

#Parent and child form mandatory fields
```

```
Roles.RoleName.Required=true

#Date format
SystemDateFormat=ddmmyy

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

B.2 Sample Schema File for Currency

This is a snippet of a schema file for reconciling currency.

```
#Schema file for ACME Currency Entitlement

#List of fields
FieldNames=currency_code,currency_name

#Unique ID Attribute
UidAttribute=currency_code

#Account Name attribute
NameAttribute=currency_name
```

B.3 Sample Schema File for Groups

This is a snippet of a schema file for reconciling groups.

```
#Schema file for ACME Group Entitlement

#List of fields
FieldNames=grpId,grpName

#Unique ID Attribute
UidAttribute=grpId

#Account Name attribute
NameAttribute=grpName
```

B.4 Sample Schema File for Roles

This is a snippet of a schema file for reconciling roles.

```
#Schema file for ACME Role Entitlement

#List of fields
FieldNames=roleId,roleName

#Unique ID Attribute
UidAttribute=roleId

#Account Name attribute
NameAttribute=roleName
```


C

Sample Groovy File

This is the snippet of the groovy file.

```
/*
 * Run like:
 * In Windows: FlatFileGenerator.cmd ..\resources\FlatFileConfiguration.groovy
trusted
 * In Linux/Unix: sh FlatFileGenerator.sh ../resources/
FlatFileConfiguration.groovy trusted
*/
trusted {
    /*
     * ITResource name
     */
    itResourceDefName='Oracle HRMS' // This will be used as a base name for all
metadata across the connector
    // itResourceName="$itResourceDefName" //the same as itResourceDefName by
default

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

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

    /*
     * Location of the flat file bundle jar
     */
    bundleJar='../..../bundle/org.identityconnectors.flatfile-1.0.1115.jar'

    /*
     * The Configuration used to run the flat file bundle mentioned above,
and get the schema by calling its SchemaOp, which is required for generating
metadata
     */
    config = [
        'schemaFile' : '/scratch/acme/flat_file/schema.properties'
```

```

]

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

dateAttributeList = ["JoiningDate"]

/**
 * Alias are used to map the OIM User Form attributes with the Connector
Attributes.
 * The Format is of 'Connector Attribute':'OIM User Form Attribute'
 * Mandatory alias shouldn't be removed. Customer can update these
mandatory attributes but should not be removed
 * Customer can add other aliases to the OIM User form fields
**/

// Some of the OIM User attributes are not used with the same display name
internally. So here are the list of attributes that need to be mapped with the
internal name then display name

/**
  Display Name  Internally Used Name
  -----
  Organization  Organization Name
  Manager       Manager Login
  E-mail        Email
*/
alias = ['__NAME__':'User Login', '__UID__':'Display
Name','Organization':'Organization Name', 'Xellerate Type':'Xellerate Type',
'Role':'Role']

// non-mandatory alias
alias += ['__ENABLE__':'Status', 'FirstName':'First Name', 'email':'Email',
'JoiningDate':'Start Date']
}

/*
 * Run like:
 * In Windows: FlatFileGenerator.cmd ..\resources\FlatFileConfiguration.groovy
target
 * In Linux/Unix: sh FlatFileGenerator.sh ../resources/
FlatFileConfiguration.groovy target
*/
target {
  /*
   * ITResource name
   */
  itResourceDefName='Flat File Target' // This will be used as a base name for
all metadata across the connector
  // itResourceName="$itResourceDefName" //the same as itResourceDefName by
default

  /**
   * Give the name of the Application Instance that needs to be generated. By
default Application Instance name is taken as itResourceDefName

```

```

    * Application Instance is not a mandatory field
    **/
    //applicationInstanceName="${itResourceDefName}"

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

    /*
    * Location of the flat file bundle jar
    */
    bundleJar='../..../bundle/org.identityconnectors.flatfile-1.0.1115.jar'

    /*
    * The Configuration used to run the flat file bundle mentioned above, and
get the schema by calling its SchemaOp, which is required for generating metadata
    */
    config = [
        'schemaFile' : '/scratch/acme/flat_file/schema.properties'
    ]

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

    lookupAttributeList=['Currency']

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

    entitlementAttributeList=["MyROLES.RoleName", "Groups" ]

    /**
    * Provide the attribute list that need to be handled as Date on process

```

```

form
    * Make sure these fields datatype in schema should be long
    * dateAttributeList is not a mandatory field
    **/

    dateAttributeList = ["JoiningDate", "MyROLES.Start_Date",
" MyROLES.End_Date"]

    /*
    * Target attribute to user fields alias
    */
    //Mandatory alias
    alias = ['__UID__':'UID', '__NAME__':'UserId']
    //Optional aliases if any (Can also be used to give short names for lengthy
attribute names)
    //alias += ['USERROLERELATIONSHIP':'USRROL', 'Comments':'Description',
'Family Name':'Last Name', 'Visibility':'Status']

    /*
    * Generate prepopulate adapters
    */
    //prepopulate = ['__NAME__':'User Login', 'FIRST_NAME':'First Name',
'LAST_NAME':'Last Name', '__PASSWORD__':'Password']
}

/*
* Run like:
* In Windows: FlatFileGenerator.cmd ..\resources\FlatFileConfiguration.groovy
disconnected
* In Linux/Unix: sh FlatFileGenerator.sh ../resources/
FlatFileConfiguration.groovy disconnected
*/
disconnected {
    /*
    * ITResource name
    */
    itResourceDefName='Access Badges' // This will be used as a base name for
all metadata across the connector
    // itResourceName="$itResourceDefName" //the same as itResourceDefName by
default

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

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

```

```

version

/*
 * Disconnected resource Flag.
 * if true, generate the disconnected metadata.
 */
disconnectedResource=true

/*
 * Location of the flat file bundle jar
 */
bundleJar='.././bundle/org.identityconnectors.flatfile-1.0.1115.jar'

/*
 * The Configuration used to run the flat file bundle mentioned above, and
get the schema by calling its SchemaOp, which is required for generating metadata
 */
config = [
    'schemaFile' : '/scratch/acme/flat_file/schema.properties'
]

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

lookupAttributeList=['Currency']

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

entitlementAttributeList=["MyROLES.RoleName", "Groups"]

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

dateAttributeList = ["JoiningDate", "MyROLES.Start_Date",
"MyROLES.End_Date"]

/*
 * Target attribute to user fields alias

```

```
    */
    //Mandatory alias
    alias = ['__UID__':'UID', '__NAME__':'UserId']
    //Optional aliases if any (Can also be used to give short names for lengthy
attribute names)
    //alias += ['USERROLERELATIONSHIP':'USRROL', 'comments':'Description',
'Family Name':'Last Name', 'Visibility':'Status']

    /*
    * Generate prepopulate adapters
    */
    prepopulate = ['__NAME__':'User Login', 'FirstName':'First Name',
'LastName':'Last Name']
}
```

D

Files and Directories in the Flat File Installation Media

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

Table D-1 Files and Directories on the Installation Media

File in the Installation Media Directory	Description
bundle/ org.identityconnectors.flatfile-1.0.1115.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	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	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	This file contains the default logging configurations of the metadata generation utility.
metadata-generator/lib/FlatFile-oim-integration.jar	This JAR file contains the class files of the FlatFile generator.
metadata-generator/lib/connector-framework-internal.jar	This JAR files contains class files that implement ICF.
metadata-generator/lib/connector-framework.jar	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/groovy-all.jar	This JAR file contains the groovy libraries required for running the FlatFile generator.
metadata-generator/resources/ FlatFileConfiguration.groovy	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. See Understanding Entries in the FlatFileConfiguration.groovy File for more information.
xml/FlatFile-ConnectorConfig.xml	This XML file contains definitions for the following connector components: <ul style="list-style-type: none">• IT resource types• IT resource instance• Lookup definitions• Scheduled jobs

E

Scheduled Jobs for Lookup Fields and Entitlement Synchronization, and Reconciliation

These are scheduled jobs that you must configure for lookup field and entitlement synchronization, and reconciliation.

Table E-1 Scheduled Jobs for Lookup Fields and Entitlement Synchronization, and Reconciliation

Scheduled Task	Description
Flat File Entitlements Loader	This scheduled job is used to load lookups and entitlements from a flat file. See Scheduled Jobs for Lookup Field and Entitlement Synchronization for information about this scheduled job.
<i>IT_RES_NAME</i> <i>FIELD_NAME</i> Loader	This scheduled job is used to load lookup values from the flat file into corresponding lookup fields in Oracle Identity Manager. This schedule job is used only if you have created the connector by using the metadata generation utility. See Scheduled Jobs for Lookup Field and Entitlement Synchronization for information about this scheduled job.
<i>IT_RES_NAME</i> <i>FIELD_NAME</i> Entitlement Loader	This scheduled job is used to synchronize entitlements with the catalog. This scheduled job is used only if you have created the connector by using the metadata generation utility. See Scheduled Jobs for Lookup Field and Entitlement Synchronization for information about this scheduled job.
Flat File Users Loader and <i>IT_RES_NAME</i> Flat File User Loader	This scheduled job is used to reconcile users from a flat file and create corresponding users in Oracle Identity Manager. See Flat File Users Loader and <i>IT_RES_NAME</i> Flat File Users Loader for information about this scheduled job.
Flat File Users Diff Sync Reconciliation and <i>IT_RES_NAME</i> Flat File Users Delete Diff Reconciliation	This scheduled job is used to perform diff-based reconciliation for users. See Flat File Users Diff Sync Reconciliation and <i>IT_RES_NAME</i> Flat File Users Delete Diff Reconciliation for information about this scheduled job.
Flat File Users Delete Reconciliation and <i>IT_RES_NAME</i> Flat File Users Delete Reconciliation	This scheduled job is used to perform a full or incremental user delete reconciliation run. See Flat File Users Delete Reconciliation and <i>IT_RES_NAME</i> Flat File Users Delete Reconciliation for information about this scheduled job.
Flat File Users Delete Sync Reconciliation <i>IT_RES_NAME</i> Flat File Delete Sync Reconciliation	This scheduled job is used to reconcile data about deleted users from a flat file containing only a list of deleted users. See Flat File Users Delete Sync Reconciliation and <i>IT_RES_NAME</i> Flat File Users Delete Sync Reconciliation for information about this scheduled job.
Flat File Accounts Loader and <i>IT_RES_NAME</i> Flat File Account Loader	This scheduled job is used to reconcile accounts from a flat file create corresponding accounts in Oracle Identity Manager. See Flat File Accounts Loader and <i>IT_RES_NAME</i> Flat File Accounts Loader for information about this scheduled job.

Table E-1 (Cont.) Scheduled Jobs for Lookup Fields and Entitlement Synchronization, and Reconciliation

Scheduled Task	Description
Flat File Accounts Diff Sync Reconciliation and <i>IT_RES_NAME</i> Flat File Accounts Delete Diff Reconciliation	This scheduled job is used to perform diff-based reconciliation for accounts. See Flat File Accounts Diff Sync Reconciliation and <i>IT_RES_NAME</i> Flat File Accounts Delete Diff Reconciliation for information about this scheduled job.
Flat File Accounts Delete Reconciliation and <i>IT_RES_NAME</i> Flat File Accounts Delete Reconciliation	This scheduled job is used to perform a full or incremental delete reconciliation run for accounts. See Flat File Accounts Delete Reconciliation and <i>IT_RES_NAME</i> Flat File Accounts Delete Reconciliation for information about this scheduled job.
Flat File Accounts Delete Sync Reconciliation and <i>IT_RES_NAME</i> Flat File Accounts Delete Sync Reconciliation	This scheduled job is used to reconcile data about deleted accounts from a flat file containing only a list of deleted accounts. See Flat File Accounts Delete Sync Reconciliation and <i>IT_RES_NAME</i> Flat File Accounts Delete Sync Reconciliation for information about this scheduled job.

Index

A

accounts
 reconciliation, [3-12](#)
adding
 new attributes, [4-10](#)
 new multivalued attributes, [4-12](#)
additional configuration entries
 commentCharacter, [2-34](#)
 customConfigParams, [2-33](#)
 encoding, [2-34](#)
 parserClassName, [2-33](#)
 postProcessClassName, [2-34](#)
 preProcessClassName, [2-34](#)
 Process Recon Event Batch Size, [2-34](#)
 Process Recon Event Thread Pool Size, [2-34](#)
 progressCheckPoint, [2-34](#)
architecture, [1-2](#)
archive, [3-20](#)

B

bat scripts
 ConnectorServer.bat, [2-28](#)
batch size entry, [2-34](#)

C

certified components, [1-1](#)
certified languages, [1-2](#)
clearing server cache, [2-32](#)
clones, [4-18](#)
cloning connector, [4-18](#)
comment character, [2-34](#)
comment characters, [2-38](#)
 configuring, [2-38](#)
complex multivalued data, [3-19](#)
 reconciling, [3-19](#)
configuring
 archival, [3-20](#)
 comment characters, [2-38](#)
 custom parser, [4-1](#)
 delimiters, [3-21](#)
 fault handling, [3-20](#)
 IT resources, [2-23](#)

 configuring (*continued*)
 preprocess and postprocess tasks, [4-6](#)
 transformation
 for reconciliation, [4-16](#)
 validation
 for reconciliation, [4-14](#)
configuring archival, [3-20](#)
configuring delimiters, [3-21](#)
connected resource, [1-5](#)
connector architecture, [1-2](#)
connector clones, [4-18](#)
connector server
 installing, [2-26](#)
 uninstalling, [2-28](#)
connector, copies, [4-18](#)
ConnectorServer.bat, [2-28](#)
ConnectorServer.properties file, [2-27](#)
copies of connector, [4-18](#)
creating
 preprocess and postprocess tasks, [4-6](#)
 sandbox, [2-30](#)
creating scheduled jobs, [3-18](#)
custom parser, [4-1](#)
 applying filters, [4-3](#)
 creating, [4-1](#)
 handling multivalued attributes, [4-4](#)
 integrating, [4-4](#)
 mandatory attributes, [4-3](#)
customizing
 reconciliation, [4-12](#)

D

defining
 scheduled jobs, [3-18](#)
delimiters, [3-21](#)

E

enabling logging, [2-34](#)
entitlement synchronization, [3-2](#)

F

fault handling, [3-20](#)

filtered reconciliation, [3-18](#)
 first-time reconciliation, [3-1](#)
 Flat File IT Resource, [2-23](#)
 FlatFileConfiguration.groovy File entries
 alias, [2-12](#)
 alias +, [2-12](#)
 bundleJar, [2-10](#)
 config, [2-10](#)
 configFile, [2-10](#)
 connectorDir, [2-9](#)
 dateAttributeList, [2-12](#)
 disconnectedResource, [2-9](#)
 itResourceDefName, [2-8](#)
 itResourceName, [2-9](#)
 prepopulate, [2-15](#)
 propertiesFile, [2-10](#)
 trusted, [2-9](#)
 version, [2-10](#)
 xmlFile, [2-9](#)

I

installing
 connector for connected resource, [2-7](#)
 connector on Oracle Identity Manager, [2-22](#)
 connector server, [2-26](#)
 integrating
 preprocess and postprocess tasks, [4-8](#)
 IT resources
 connector server, [2-39](#)
 Flat File Accounts, [2-24](#)
 Flat File Entitlements, [2-23](#)
 Flat File IT Resource, [2-23](#)
 Flat File Users, [2-24](#)

J

Java Connector Server
 installing, [2-26](#)
 uninstalling, [2-28](#)

L

limited reconciliation, [3-18](#)
 logging enabling, [2-34](#)
 lookup definitions
 Lookup.FlatFile.Configuration, [1-9](#)
 Lookup.FlatFile.Configuration.Trusted, [1-10](#)
 Lookup.FlatFile.EntFieldMap, [1-11](#)
 Lookup.FlatFile.UM.Configuration, [1-11](#)
 Lookup.FlatFile.UM.Configuration.Trusted,
 [1-12](#)
 lookup field, [3-2](#)
 lookup field synchronization, [3-2](#)

M

mapping
 resource object reconciliation fields, [2-18](#)

P

preprocess and postprocess tasks, [4-6](#)
 creating, [4-6](#)
 integrating, [4-8](#)

R

reconciliation
 configuring
 scheduled jobs, [3-18](#)
 customizing, [4-10](#)
 first time, [3-1](#)
 scheduled jobs, [3-4](#)
 reconciling
 accounts, [3-12](#)
 complex multivalued data, [3-19](#)
 users, [3-6](#)

S

sample
 schema file
 for roles, [B-2](#)
 for users, [B-1](#)
 sandbox, [2-30](#)
 activating, [2-30](#)
 creating, [2-30](#)
 publishing, [2-31](#)
 scheduled job attributes, [3-5](#)
 scheduled jobs, [3-4](#)
 attributes, [3-5](#)
 defining, [3-18](#)
 Flat File Accounts Delete Reconciliation,
 IT_RES_NAME Flat File Accounts
 Delete Reconciliation, [3-15](#)
 Flat File Accounts Delete Sync
 Reconciliation, IT_RES_NAME Flat
 File Accounts Delete Sync
 Reconciliation, [3-16](#)
 Flat File Accounts Diff Sync Reconciliation,
 IT_RES_NAME Flat File Accounts
 Delete Diff Reconciliation, [3-13](#)
 Flat File Accounts Loader, IT_RES_NAME
 Flat File Accounts Loader, [3-12](#)
 Flat File Users Delete Reconciliation,
 IT_RES_NAME Flat File Users
 Delete Reconciliation, [3-9](#)

- scheduled jobs (*continued*)
 - Flat File Users Delete Sync Reconciliation, IT_RES_NAME Flat File Users Delete Sync Reconciliation, [3-10](#)
 - Flat File Users Diff Sync Reconciliation, IT_RES_NAME Flat File Users Delete Diff Reconciliation, [3-8](#)
 - Flat File Users Loader, IT_RES_NAME Flat File Users Loader, [3-6](#)
 - schema file
 - format, [2-6](#)
 - schema file qualifiers
 - DataType, [2-3](#)
 - EmbeddedObjectClass, [2-3](#)
 - False, [2-5](#)
 - FieldNames, [2-2](#)
 - Multivalued, [2-3](#)
 - NameAttribute, [2-2](#)
 - PasswordAttribute, [2-2](#)
 - Required, [2-3](#)
 - StatusAttribute, [2-2](#)
 - Subfields, [2-3](#)
 - SystemDateFormat, [2-2](#)
 - True, [2-3](#)
 - UidAttribute, [2-2](#)
 - server cache, clearing, [2-32](#)
 - stages of connector deployment
 - installation, [2-22](#)
 - stages of connector deployment (*continued*)
 - postinstallation, [2-29](#)
 - preinstallation, [2-1](#)
 - supported
 - connector server versions, [1-2](#)
 - flat file formats, [1-2](#)
 - releases of Oracle Identity Manager, [1-2](#)
 - target systems, [1-2](#)
 - synchronization
 - entitlement, [3-2](#)
-
- ## T
-
- target system, multiple installations, [4-18](#)
 - thread pool size entry, [2-34](#)
 - transformation, [4-16](#)
-
- ## U
-
- uninstalling connector server, [2-28](#)
 - update jars, [4-9](#)
 - users
 - reconciliation, [3-6](#)
-
- ## V
-
- validation, [4-14](#)