Oracle® Identity Governance

Configuring the UNIX Application

12c (12.2.1.3.0)

F12379-05

August 2020

Oracle Identity Governance Configuring the UNIX Application, 12c (12.2.1.3.0)

F12379-05

Primary Author: Alankrita Prakash

Contributors: Gowri.G.R, Uday Tripathi

Contents

Preface

What's New in this Guide?

1 About the UNIX Connector

2 Creating an Application By Using the UNIX Connector

3 Configuring the UNIX Connector

4 Performing the Postconfiguration Tasks for the UNIX Connector

5 Using the UNIX Connector

6 Extending the Functionality of the UNIX Connector

7 Upgrading the UNIX Connector

8 Testing and Troubleshooting the UNIX Connector

List of Figures

List of Tables

Preface

This guide describes the connector that is used to onboard UNIX applications to Oracle Identity Governance.

Audience

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

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

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

`http://docs.oracle.com/middleware/12213/oig/index.html`

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

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

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

`http://docs.oracle.com/middleware/oig-connectors-12213/index.html`

For information about Oracle Identity Manager Connectors 11.1.1 documentation, visit the following Oracle Help Center page:

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

Conventions

The following text conventions are used in this document:

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

What's New in this Guide?

These are the updates made to the software and documentation for release 12.2.1.3.0 of Configuring the UNIX Application.

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

- Software Updates

  These include updates made to the connector software.
- Documentation-Specific Updates

  These include major changes made to the connector documentation. These changes are not related to software updates.

Software Updates

These are the updates made to the connector software.

Software Updates in Release 12.2.1.3.0

The following is the software update in release 12.2.1.3.0:

Support for Onboarding Applications Using the Connector

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

Documentation-Specific Updates

These are the updates made to the connector documentation.

Documentation-Specific Updates in Release 12.2.1.3.0

The following documentation-specific update has been made in revision "05" of this guide:

The "Target Systems" row of Table 1-1 has been updated to include IBM AIX 7.2 version.

The following documentation-specific update has been made in revision "04" of this guide:

Information about Oracle Identity Manager versions prior to 11g Release 2 PS3 (11.1.2.3.0) has been removed from the guide.

The following documentation-specific update has been made in revision "03" of this guide:

The "Oracle Identity Governance or Oracle Identity Manager" row of Table 1-1 has been updated to include support for Oracle Identity Governance 12*c* (12.2.1.4.0).

The following documentation-specific updates have been made in revision "02" of this guide:

- The "Target Systems" row in Table 1-1 has been modified to include the supported version Oracle Enterprise Linux 6.*x* and 7.*x.*
- A "Note" regarding sudo privileges has been added to Configuring SSH Public Key Authentication for Solaris, Configuring SSH Public Key Authentication for HP-UX, and Configuring SSH Public Key Authentication for AIX.
- A prerequisite step has been removed from Creating a Target System SUDO User Account for Connector Operations.
- Several broken links were fixed throughout the document.

The following documentation-specific update has been made in revision "01" of this guide:

This is the first release of the Oracle Identity Governance Connector for UNIX. Therefore, there are no documentation-specific updates in this release.

1 About the UNIX Connector

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

The UNIX connector lets you onboard applications in Oracle Identity Governance for UNIX-based target systems using SSH or Telnet protocol.

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

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

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

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

- Certified Components
- Usage Recommendation
- Certified Languages
- Supported Connector Operations
- Connector Architecture
- Supported Connector Features Matrix
- Connector Features

Note:

In this guide, the term *Oracle Identity Governance server* refers to the computer on which Oracle Identity Governance is installed.

1.1 Certified Components

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

Note:

If you are using Oracle Identity Manager release 11.1.*x*, then you can install and use the connector only in the CI-based mode. If you want to use the AOB application, then you must upgrade to Oracle Identity Governance release 12.2.1.3.0 or later.

Table 1-1 Certified Components

Component	Requirement for AOB Application	Requirement for CI-Based Connector
Oracle Identity Governance or Oracle Identity Manager	You can use one of the following releases of Oracle Identity Manager or Oracle Identity Governance: • Oracle Identity Governance 12*c* (12.2.1.4.0) • Oracle Identity Governance 12*c* (12.2.1.3.0)	You can use one of the following releases of Oracle Identity Manager or Oracle Identity Governance: • Oracle Identity Governance 12*c* (12.2.1.4.0) • Oracle Identity Governance 12*c* (12.2.1.3.0) **Note:** If you are using Oracle Identity Governance 12*c* (12.2.1.3.0), then download and apply the patch 26616250 from My Oracle Support. Failing to apply this patch causes target resource user reconciliation runs to fail. • Oracle Identity Manager 11*g* Release 2 PS3 (11.1.2.3.0)
Oracle Identity Governance or Oracle Identity Manager JDK	JDK 1.7 or later	JDK 1.7 or later

Table 1-1 (Cont.) Certified Components

Component	Requirement for AOB Application	Requirement for CI-Based Connector
Target systems	The target system can be any one of the following operating systems: • HP-UX 11.11, 11.20, 11.31 • IBM AIX 5L Version 5.2, 5.3, 6.1, 7.1, 7.2 • Oracle Enterprise Linux 5.2+(32-bit) and 64-bit versions of 5.2, 5.3, 5.4, 5.5, 5.6. 6.*x*, 7.*x* • Red Hat Enterprise Linux AS 2.1, 3, 4.*x* • Red Hat Enterprise Linux ES 3, 4.*x* • Red Hat Linux 5.5+ Enterprise Edition (64-bit) • Solaris 9, 10, 11 **Note:** You can also configure and use the connector on any other UNIX-based operating system that supports SSH and Telnet protocols. See Configuring the Connector for a New Target System for more information.	The target system can be any one of the following operating systems: • HP-UX 11.11, 11.20, 11.31 • IBM AIX 5L Version 5.2, 5.3, 6.1, 7.1, 7.2 • Oracle Enterprise Linux 5.2+(32-bit) and 64-bit versions of 5.2, 5.3, 5.4, 5.5, 5.6, 6.*x*, 7.*x* • Red Hat Enterprise Linux AS 2.1, 3, 4.*x* • Red Hat Enterprise Linux ES 3, 4.*x* • Red Hat Linux 5.5+ Enterprise Edition (64-bit) • Solaris 9, 10, 11 **Note:** You can also configure and use the connector on any other UNIX-based operating system that supports SSH and Telnet protocols. See Configuring the Connector for a New Target System for more information.
Connector Server	11.1.2.1.0	11.1.2.1.0
Connector Server JDK	JDK 1.7 or later	JDK 1.7 or later
Other systems	OpenSSH, OpenSSL, operating system patches (HP-UX), and SUDO software (only if the SUDO Admin mode is required)	OpenSSH, OpenSSL, operating system patches (HP-UX), and SUDO software (only if the SUDO Admin mode is required)

Table 1-1 (Cont.) Certified Components

Component	Requirement for AOB Application	Requirement for CI-Based Connector
Target system user account	Depending on the target system that you are using, the target system user account can be one of the following: • For AIX, HP-UX, and Linux environments: root user or sudo user • For Solaris: root user, sudo user, RBAC user You provide the credentials of this user account as part of Basic Configuration Parameters while creating an application.	Depending on the target system that you are using, the target system user account can be one of the following: • For AIX, HP-UX, and Linux environments: root user or sudo user • For Solaris: root user, sudo user, RBAC user You provide the credentials of this user account while configuring the IT resource.
Character encoding supported by the target system	The target system must support the default C (POSIX) locale. Use the following command to check the locale that the target system supports: `locale -a`	The target system must support the default C (POSIX) locale. Use the following command to check the locale that the target system supports: `locale -a`

> **Note:**
>
> The connector requires sh shell on the target system to run the scripts. Therefore, the connector switches to sh before running the commands.
>
> If the user account indicated in the loginUser basic configuration parameter has access to sh and the user account can switch to sh, then there is no restriction on the original login shell.

1.2 Usage Recommendation

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

- If you are using Oracle Identity Governance 12c (12.2.1.3.0), then use the latest 12.2.1.*x* version of this connector. Deploy the connector using the **Applications** option on the **Manage** tab of Identity Self Service.
- If you are using any of the Oracle Identity Manager releases listed in the “Requirement for CI-Based Connector” column in Certified Components, then use

the 11.1.*x* version of the connector. If you want to use the 12.2.1.*x* version of this connector, then you can install and use it only in the CI-based mode. If you want to use the AOB application, then you must upgrade to Oracle Identity Governance release 12.2.1.3.0.

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

1.3 Certified Languages

These are the languages that the connector supports.

- Arabic
- Chinese (Simplified)
- Chinese (Traditional)
- Czech
- Danish
- Dutch
- English
- Finnish
- French
- French (Canadian)
- German
- Greek
- Hebrew
- Hungarian
- Italian
- Japanese
- Korean
- Norwegian
- Polish
- Portuguese
- Portuguese (Brazilian)
- Romanian
- Russian
- Slovak

- Spanish
- Swedish
- Thai
- Turkish

1.4 Supported Connector Operations

These are the list of operations that the connector supports for your target system.

Table 1-2 Supported Connector Operations

Operation	Supported?
User Management	
Create user	Yes
Update user	Yes
Delete user	Yes
Enable user	Yes
Disable user	Yes
Update user login	Yes
Update user shell	Yes
Update UID	Yes
Group Management	
Update primary group	Yes
Insert secondary group	Yes
Update secondary group	Yes
Delete secondary group	Yes
Entitlement Grant Management	
Add role	Yes
Update GECOS	Yes
Update home directory	Yes
Update inactive days	Yes
Update expire date	Yes
Update password	Yes

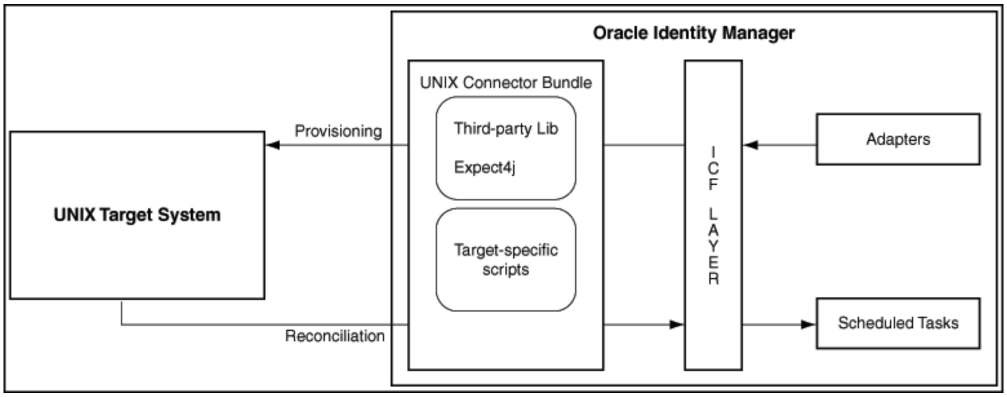
1.5 Connector Architecture

You can configure the UNIX connector to run in the Target (or account management) and Authoritative (or trusted) mode, and is implemented using the Integrated Common Framework (ICF) component.

The ICF is a component that provides basic reconciliation and provisioning operations that are common to all Oracle Identity Governance connectors. In addition, ICF provides common features that developers would otherwise need to implement on their own, such as connection pooling, buffering, time outs, and filtering. The ICF is shipped along with Oracle Identity Governance. Therefore, you need not configure or modify the ICF.

This connector enables management of target system accounts through Oracle Identity Governance. Figure 1-1 shows the architecture of the connector.

Figure 1-1 Architecture of the UNIX Connector



As shown in this figure, the UNIX connector enables you to use the target system as a managed resource (target) or as an authoritative (trusted) source of identity data for Oracle Identity Governance.

In the target mode of the connector, information about users created or modified directly on the target system can be reconciled into Oracle Identity Governance. In addition, you can use Oracle Identity Governance to perform provisioning operations on the target system.

In the authoritative configuration of the connector, users are created or modified only on the target system and information about these users is reconciled into Oracle Identity Governance.

Provisioning involves creating and managing user accounts. When you allocate (or provision) a UNIX resource to an OIM User, the operation results in the creation of an account on the target system for that user. Similarly, when you update the resource on Oracle Identity Governance, the same update is made to the account on the target system.

During provisioning operations, adapters carry provisioning data submitted through the process form to the Expect4j third-party library, which in turn submits the provisioning data to the target system. The user account maintenance commands accept provisioning data from the adapters, carry out the required operation on the target system, and return the response from the target system to the adapters. The adapters return the response to Oracle Identity Governance.

1.6 Supported Connector Features Matrix

Provides the list of features supported by the AOB application and CI-based connector.

Table 1-3 Supported Connector Features Matrix

Feature	AOB Application	CI-Based Connector
Switch between SSH and Telnet protocols to connect to UNIX-based target systems	Yes	Yes
Run scripts on a computer where the UNIX connector is deployed	Yes	Yes
Configure the connector to support an additional flavor of UNIX by using custom scripts	Yes	Yes
Support multiple instances and multiple versions of UNIX	Yes	Yes
Integrate the target system as a target resource and an authoritative source of Oracle Identity Governance	Yes	Yes
Perform full and incremental reconciliation	Yes	Yes
Perform limited reconciliation	Yes	Yes
Perform batched reconciliation	Yes	Yes
Configure validation and transformation of account data	Yes	Yes
Use connector server	Yes	Yes
Reconcile user account status information from the target system	Yes	Yes
Add custom attributes for reconciliation and provisioning	Yes	Yes

1.7 Connector Features

The features of the connector include support for connector server, target resource and trusted source reconciliation, configuring custom scripts to support additional flavors of UNIX, reconciliation of all existing or modified account data, limited and batched reconciliation, transformation and validation of account data during reconciliation and provisioning, and so on.

The following are the features of this connector:

- Support for Switching Between SSH and Telnet Protocols
- Support for Running Custom Scripts
- Support for Configuring the Connector for a New Target System
- Support for Multiple Instances and Multiple Versions of UNIX
- Support for Both Target Resource and Trusted Source Reconciliation
- Support for Both Full and Incremental Reconciliation
- Support for Limited Reconciliation

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

1.7.1 Support for Switching Between SSH and Telnet Protocols

You can switch between SSH and Telnet protocols to connect to UNIX-based target systems. You can specify the connection type by using the connectionType parameter of the IT Resource.

The connector supports the following connection types:

- SSH - This is the default connection. Used for SSH with password-based authentication.
- SSHPUBKEY - Used for SSH with key-based authentication.
- TELNET - Used for Telnet connection.

See Basic Configuration Parameters for related information.

1.7.2 Support for Running Custom Scripts

You can run scripts on a computer where the UNIX connector is deployed. You can configure custom scripts to support additional flavors of UNIX.

You can configure the scripts to run before or after the create, update, or delete an account provisioning operations. For example, you could configure a script to run before a user is created by the connector. See Configuring Action Scripts for more information.

1.7.3 Support for Configuring the Connector for a New Target System

You can configure the connector to support an additional flavor of UNIX by using custom scripts.

By default, the connector uses pre-configured scripts to support AIX, HP-UX, Linux, and Solaris. You can customize these scripts to support an additional flavor of UNIX. See Configuring the Connector for a New Target System for more information.

1.7.4 Support for Multiple Instances and Multiple Versions of UNIX

The connector supports multiple instances and multiple versions of UNIX.

You can deploy a single connector bundle on Oracle Identity Governance and create multiple instances and multiple versions of UNIX. Then, you can use Oracle Identity Governance to manage accounts on these target systems. See Configuring the Connector for Multiple Installations of the Target System for more information.

1.7.5 Support for Both Target Resource and Trusted Source Reconciliation

You can use the connector to configure the target system as either a target resource or trusted source of Oracle Identity Governance.

See Configuring Reconciliation for more information.

1.7.6 Support for Both Full and Incremental Reconciliation

After you create the application, you can perform full reconciliation to bring all existing user data from the target system to Oracle Identity Governance. After the first full reconciliation run, incremental reconciliation is automatically enabled from the next run of the user reconciliation.

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

1.7.7 Support for Limited Reconciliation

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

See Performing Limited Reconciliation for more information.

1.7.8 Support for Batched Reconciliation

You can break down a reconciliation run into batches by specifying the number of records that must be included in each batch.

See Performing Batched Reconciliation for more information.

1.7.9 Support for the Connector Server

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

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

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

1.7.10 Transformation and Validation of Account Data

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

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

2 Creating an Application By Using the UNIX Connector

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

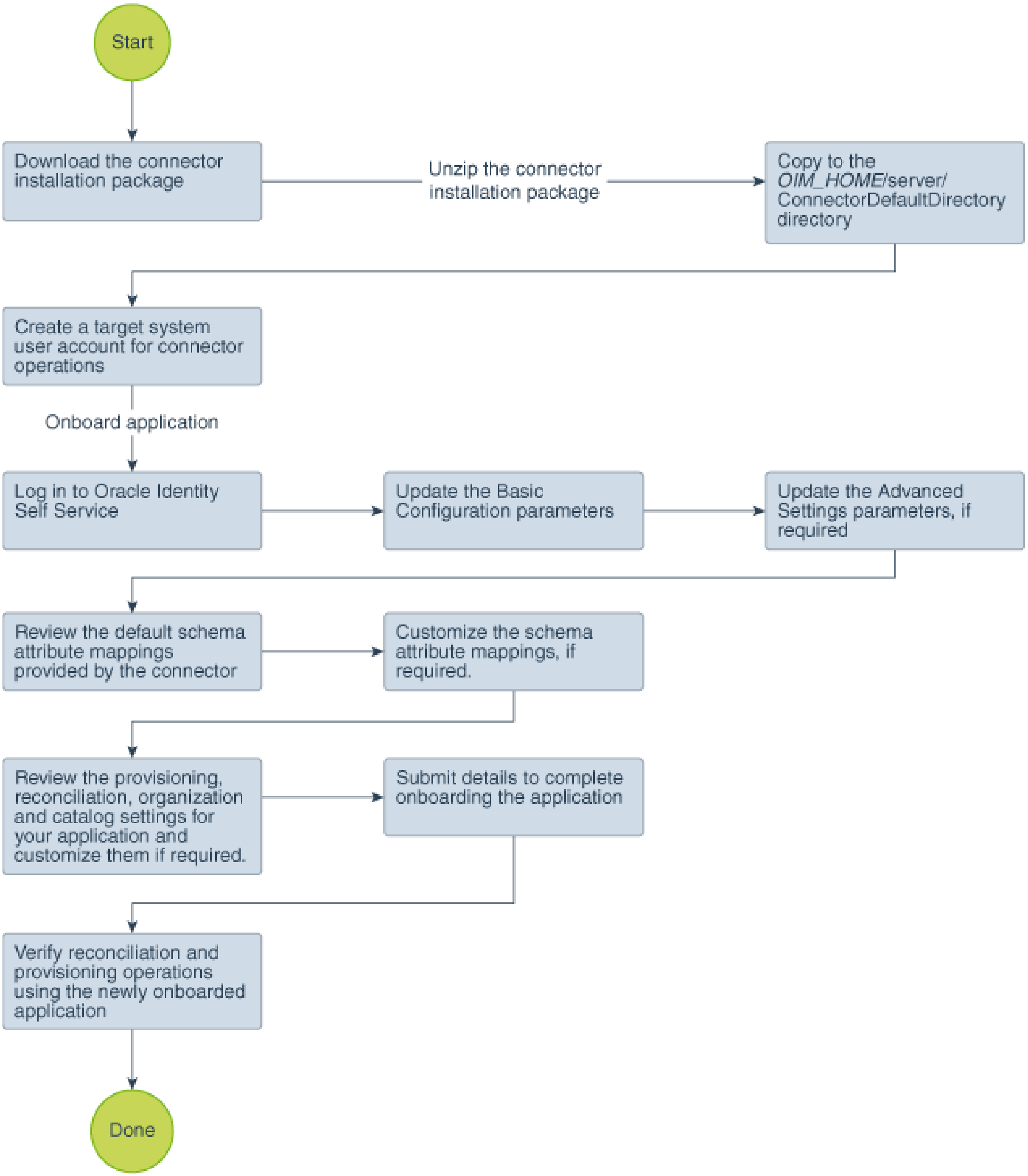
- Process Flow for Creating an Application By Using the Connector
- Prerequisites for Creating an Application By Using the Connector
- Creating an Application By Using the Connector

2.1 Process Flow for Creating an Application By Using the Connector

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

Figure 2-1 is a flowchart depicting high-level steps for creating an application in Oracle Identity Governance by using the connector installation package.

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


Start
Download the connector installation package
Unzip the connector installation package
Copy to the *OIM_HOME*/server/ ConnectorDefaultDirectory directory
Create a target system user account for connector operations
Onboard application
Log in to Oracle Identity Self Service
Update the Basic Configuration parameters
Update the Advanced Settings parameters, if required
Review the default schema attribute mappings provided by the connector
Customize the schema attribute mappings, if required.
Review the provisioning, reconciliation, organization and catalog settings for your application and customize them if required.
Submit details to complete onboarding the application
Verify reconciliation and provisioning operations using the newly onboarded application
Done

2.2 Prerequisites for Creating an Application By Using the Connector

Learn about the tasks that you must complete before you create the application.

- Downloading the Connector Installation Package
- Configuring the Target System

2.2.1 Downloading the Connector Installation Package

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

To download the connector installation package:

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

   You must accept the license agreement before you can download the installation package.
4. Download and save the installation package to any directory on the computer hosting Oracle Identity Governance.
5. Extract the contents of the installation package to any directory on the computer hosting Oracle Identity Governance. This creates a directory named *CONNECTOR_NAME-RELEASE_NUMBER.* For example, GenericUnix-12.2.1.3.0
6. Copy the *CONNECTOR_NAME-RELEASE_NUMBER* directory to the *OIM_HOME*/server/ConnectorDefaultDirectory directory.

2.2.2 Configuring the Target System

Depending on the target system and your requirements, perform these procedures to configure your environment, install OpenSSH, create a target system user account with the minimum permissions required to perform connector operations, create an RBAC user account, and configure public key and SSH public key authentication.

- Configuring Solaris, Linux, and AIX
- Configuring HP-UX
- Installing OpenSSH
- Creating a Target System SUDO User Account for Connector Operations
- Creating an RBAC User Account for Connector Operations on Solaris
- Configuring Public Key Authentication
- Configuring SSH Public Key Authentication

2.2.2.1 Configuring Solaris, Linux, and AIX

Perform the following steps to configure Solaris, Linux, and AIX environments:

1. Ensure that the /etc/passwd and /etc/shadow files are available on the UNIX server.
2. Create a directory on the target system where the connector can create mirror files for the /etc/passwd and /etc/shadow files.

   This directory is specified in the Mirror Files Location parameter. The default value is `/etc/connector_mirror_files` for a Target application and `/etc/connector_mirror_files_trusted` for an Authoritative application. If the directory path is different from the default value, then you must update the parameter value. The Login User (sudo or root user) must have read and write privileges to this directory.

2.2.2.2 Configuring HP-UX

Perform the following steps for HP-UX environments:

1. If you want to switch to HP-UX Trusted mode, then:

   > **Note:**
   >
   > If you are converting the target system to the trusted system, then please make sure that no shadow file exists on the target after it is converted to trusted system. You can use `pwunconv` command to get rid of the shadow file, if it exists.

   a. Log in as root and then run the following command:

   ```
   /usr/bin/sam
   ```

   ```
   /usr/sbin/sam
   ```

   b. Select **Auditing and Security** and then select **System Security Policies.** A message is displayed asking if you want to switch to the trusted mode.

   c. Click **Yes.** The following message is displayed:

   ```
   System changed successfully to trusted system
   ```

2. Ensure that the /etc/passwd and /etc/shadow directories are available on the target server.
3. Create a directory on the target system where the connector can create mirror files for the /etc/passwd and /etc/shadow files.

   This directory is specified in the Mirror Files Location parameter. The default value is `/etc/connector_mirror_files` for a Target application and `/etc/connector_mirror_files_trusted` for an Authoritative application. If the directory path is different from the default value, then you must update the parameter value. The loginUser (sudo or root user) must have read and write privileges to this directory.

2.2.2.3 Installing OpenSSH

Perform the following procedures to install OpenSSH on the target system:

- Installing OpenSSH for Solaris 9
- Installing OpenSSH for Solaris 10 and Later Versions
- Installing OpenSSH for HP-UX
- Installing OpenSSH for Linux
- Installing OpenSSH for AIX

2.2.2.3.1 Installing OpenSSH for Solaris 9

Perform the following steps to install OpenSSH on Solaris 9:

1. If SSH is not installed on the Solaris server, then install the appropriate OpenSSH.
2. Create a group with the name `sshd` and group ID `27`. Add a user with the name `sshadmin` to this group.
3. To enable root logins, change the value of `PermitRootLogin` in the /etc/ssh/sshd_config file as follows:

   ```
   PermitRootLogin yes
   ```

   > **Note:**
   >
   > Implement this change only if it does not violate local security policies. If Public Key Authentication is enabled, then you can change the value of `PermitRootLogin` to `without-password`.
   >
   > Instead of using the root account, if you can use a user account with sudo privileges, then you do not need to perform this step.

2.2.2.3.2 Installing OpenSSH for Solaris 10 and Later Versions

By default, OpenSSH is installed on Solaris 10 and later versions. If it is not installed, then install the OpenSSH server from the operating system installation CD. To enable SSH, make the following changes in the /etc/ssh/ssh_config file:

1. Remove the comment character from the `Host *` line.
2. To enable root logins, change the value of `PermitRootLogin` in the /etc/ssh/sshd_config file as follows:

   ```
   PermitRootLogin yes
   ```

> **Note:**
>
> Implement this change only if it does not violate local security policies. If Public Key Authentication is enabled, then you can change the value of `PermitRootLogin` to `without-password`.
>
> Instead of using the root account, if you can use a user account with sudo privileges, then you do not need to perform this step.

2.2.2.3.3 Installing OpenSSH for HP-UX

If SSH is not installed on the UNIX server, then install the appropriate OpenSSH from the installation media.

2.2.2.3.4 Installing OpenSSH for Linux

By default, OpenSSH is installed on Red Hat Linux. If it is not installed, then install the OpenSSH server from the operating system installation CD.

2.2.2.3.5 Installing OpenSSH for AIX

If SSH is not installed on the AIX server, then from the installation media:

1. Install OpenSSL.
2. Install PRNG.
3. Install OpenSSH.
4. To enable root logins, change the value of `PermitRootLogin` in the /etc/ssh/sshd_config file as follows:

   ```
   PermitRootLogin yes
   ```

> **Note:**
>
> Implement this change only if it does not violate local security policies. If Public Key Authentication is enabled, then you can change the value of `PermitRootLogin` to `without-password`.
>
> Instead of using the root account, if you can use a user account with sudo privileges, then you do not need to perform this step.

2.2.2.4 Creating a Target System SUDO User Account for Connector Operations

Oracle Identity Governance uses a target system account for performing reconciliation and provisioning operations. On all supported target systems, this account must be either the root user or sudo user.

> **See Also:**
>
> Privileges Required for Performing Provisioning and Reconciliation for information about the privileges required to perform connector operations

To create a target system user account with the minimum permissions required to perform connector operations, perform the following procedure:

1. If SUDO is not installed on the target system, then install it from the installation media.
2. Use the `visudo` command to edit and customize the /etc/sudoers file according to your requirements.

   > **Note:**
   >
   > If you cannot use the `visudo` command to edit the `sudoers` file, then:
   >
   > a. Enter the following command:
   >
   > ```
   > chmod 777 /etc/sudoers
   > ```
   >
   > b. Make the required changes in the `sudoers` file.
   >
   > c. Enter the following command:
   >
   > ```
   > chmod 440 /etc/sudoers
   > ```

   For example, if you have a group named `mqm` on the Linux server and require all members of the group to act as SUDO users with all possible privileges, then the `sudoers` file must contain the following line:

   ```
   mqm ALL= (ALL) ALL
   ```

   This example is only a sample configuration. If you need other group members or individual users to be SUDO users with specific privileges, then edit this file as was done for the sample value `mqm`.

   Therefore, the SUDO user must have the privileges required to run these commands.

   > **Note:**
   >
   > `NOPASSWD: ALL` option for any SUDO user or group is supported. To configure this, you need to update the Sudo Passwd Expect Expression parameter. See Advanced Settings Parameters.

3. Edit the same `sudoers` file so that the SUDO user stays validated for 10 minutes after being validated once. You may need to increase the timeout if the reconciliation operation takes longer than 10 minutes and if you encounter errors such as "Permission denied". At the beginning of each operation, the connector validates the user using `sudo -v` option so that the operation stays validated for a

maximum of 10 minutes. After carrying out the operation, the connector runs the `sudo -k` to kill the validation.

Add the following line under the `# Defaults specification` header:

```
Defaults timestamp_timeout=10
```

This is a prerequisite for this connector to work successfully.

4. Create a SUDO user as follows:
   a. Enter the following command:

      ```
      useradd -g group_name -d /home/directory_name -m user_name
      ```

      In this command:

      `group_name` is the SUDO users group for which there is an entry in the `/etc/sudoers` file.

      `directory_name` is the name of the directory in which you want to create the default directory for the user.
   b. In the .bash_profile file, which is created in the /home/*directory_name* directory, add the following lines to set the `PATH` environment variable:

      ```
      PATH=/usr/sbin:$PATH
      export PATH
      ```
5. In the sudo user's .bashrc, .cshrc, or .kshrc file, which is created in the sudo user's home directory, add the following line to change the prompt end character from $ (dollar sign) to # (pound sign):

   ```
   PS1="[\\u@\\h:\\w]#"
   ```

   The encrypted passwords in the shadow file contain $ (dollar sign), which matches the default prompt end character. You must change the prompt end character to ensure that changes made to the shadow file are reconciled correctly.
6. Login with the sudo user.
7. Run the `sudo -k` command on the target system to clear the validation.
8. Run the `sudo -v` command on the target system and ensure that the password prompt is displayed.

   The connector would not work if the sudo user is not prompted for password at this step.

2.2.2.5 Creating an RBAC User Account for Connector Operations on Solaris

On Solaris, you can either create a sudo user or apply the role-based access control (RBAC) feature to create an account and assign to it the minimum privileges required for connector operations.

> **Note:**
>
> You use the RBAC Role Expect Expressions parameter to specify if you want to use an RBAC user. See Advanced Settings Parameters.

To create an RBAC user account:

1. Run the following command to create a role for the user.

   ```
   roleadd -d /export/home/ROLE_NAME -m ROLE_NAME
   ```

   In this command, replace *ROLE_NAME* with the name that you want to assign to the role, for example, `OIMRole`.

2. Run the following command to assign a password to the role:

   ```
   passwd ROLE_NAME
   ```

   At the prompt, enter a password for the role.

   > **See Also:**
   >
   > Privileges Required for Performing Provisioning and Reconciliation for information about the privileges required to run the commands that are used for provisioning and reconciliation

3. Create a profile for the user as follows:

   a. Open the /etc/security/prof_attr file in a text editor and insert the following line in the file:

   ```
   PROFILE_NAME:::Oracle Identity Manager Profile:
   ```

   In this line, replace *PROFILE_NAME* with the name that you want to assign to the profile, for example, `OIMProf`.

   b. Save and close the file.

4. Add execution attribute entries in the /etc/security/exec_attr file. Each entry defines a task to be run and the uid that the role will assume when running the task.

   Open the /etc/security/exec_attr file in a text editor, and insert the following lines:

   > **Note:**
   >
   > There are seven fields in this file, and the colon (:) is used as the delimiting character.
   >
   > On Solaris 10, the value `suser` can be replaced with `solaris`.
   >
   > Some of the entries contain `euid`. These instances of `euid` can be replaced with `uid`.

   ```
   PROFILE_NAME:suser:cmd:::/usr/sbin/usermod:uid=0
   PROFILE_NAME:suser:cmd:::/usr/sbin/useradd:uid=0
   PROFILE_NAME:suser:cmd:::/usr/sbin/userdel:uid=0
   PROFILE_NAME:suser:cmd:::/usr/bin/passwd:uid=0
   PROFILE_NAME:suser:cmd:::/usr/bin/cat:euid=0
   PROFILE_NAME:suser:cmd:::/usr/bin/diff:euid=0
   PROFILE_NAME:suser:cmd:::/usr/bin/sort:euid=0
   PROFILE_NAME:suser:cmd:::/usr/bin/rm:uid=0
   ```

```
PROFILE_NAME:suser:cmd:::/usr/bin/grep:euid=0
PROFILE_NAME:suser:cmd:::/usr/bin/egrep:euid=0
PROFILE_NAME:suser:cmd:::/bin/echo:euid=0
PROFILE_NAME:suser:cmd:::/bin/sed:euid=0
```

You can add similar entries for other commands if you have customized the pre-configured Solaris scripts to use other commands.

5. Run the following command to associate the profile with the role:

   ```
   rolemod -P PROFILE_NAME ROLE_NAME
   ```

6. Run the following command to create the user:

   ```
   useradd -d /export/home/USER_NAME -m USER_NAME
   ```

7. Run the following command to assign a password to the user:

   ```
   passwd USER_NAME
   ```

8. Run the following command to grant the role to the user:

   ```
   usermod -R ROLE_NAME USER_NAME
   ```

9. To verify the changes that you have made, open the /etc/user_attr file in a text editor and verity that the following entries are present in the file:

   ```
   ROLE_NAME::::type=role;profiles=PROFILE_NAME
   USER_NAME::::type=normal;roles=ROLE_NAME
   ```

2.2.2.6 Configuring Public Key Authentication

To configure Public Key Authentication:

> **Note:**
>
> - If Public Key Authentication is used, then an RBAC user for a Solaris target system cannot be used.
> - This section contains the procedure to configure Public Key Authentication for a root user. It can also be configured for a SUDO user.

1. Copy the util/privateKeyGen.sh file from the installation media directory to any directory on the target system server.
2. Open this script file in a text editor and specify a working directory path other than the default value given in the file.
3. If required, enter the following command:

   For Solaris or Linux:

   ```
   dos2unix privateKeyGen.sh privateKeyGen.sh
   ```

   For HP-UX:

   ```
   dos2ux privateKeyGen.sh
   ```

4. Run the privateKeyGen.sh script on the UNIX server.

Provide a secure passphrase when prompted. Do not leave the passphrase blank. If you do so, the connector operations will be affected.

When these commands are run, the following files are created in the $HOME/.ssh directory:

- id_rsa: This is a private key file.
- authorized_keys: This file lists public keys that can be used to log in.

5. When the keys are generated successfully, edit the sshd_config file for Public Key Authentication and test login.
6. After successfully testing login, copy the id_rsa file to the following directory:

   ```
   OIM_HOME/server/ConnectorDefaultDirectory/SSH/config
   ```

   You can also copy the file to any directory that is readable and accessible by Oracle Identity Governance. The permissions for the keys should not be changed. If you change it for copying, then you must revert the permissions.

   > **Note:**
   >
   > This release of the connector has been tested and certified only for RSA keys, and not DSA. In addition, this connector has been tested and certified for only single key configuration and not multiple keys.

2.2.2.7 Configuring SSH Public Key Authentication

Depending on the target system and your requirements, perform some of the following procedures to configure SSH Public Key Authentication:

- Configuring SSH Public Key Authentication for Solaris
- Configuring SSH Public Key Authentication for HP-UX
- Configuring SSH Public Key Authentication for Linux
- Configuring SSH Public Key Authentication for AIX

2.2.2.7.1 Configuring SSH Public Key Authentication for Solaris

Perform the following steps to configure SSH Public Key Authentication on Solaris:

1. Set the following parameters in the /etc/ssh/sshd_config file:

   ```
   PubKeyAuthorization yes
   PasswordAuthentication no
   PermitRootLogin yes
   ```

> **Note:**
>
> Change the value of `PermitRootLogin` to `yes` only if it does not violate local security policies. If Public Key Authentication is enabled, then you can change the value of `PermitRootLogin` to `without-password`.
>
> Instead of using the root account, if you can use a user account with sudo privileges, then you do not need to perform this step.

2. To restart the SSH server, enter the following commands:
   - `/etc/init.d/sshd stop`
   - `/etc/init.d/sshd start`
3. To test login:

   ```
   ssh -i /.ssh/id_rsa -l root server_IP_address
   ```

   This command prompts you for the passkey before setting up the connection.

   > **Note:**
   >
   > Instead of using the root account, you can use a user account with sudo privileges, if desired.

4. Set the `privateKey[LOADFROMURL]` advanced settings parameter to include the complete path of the `id_rsa` file with the prefix `file://`.

   For example:

   ```
   file:///OIM_HOME/server/ConnectorDefaultDirectory/SSH/config/id_rsa
   ```

2.2.2.7.2 Configuring SSH Public Key Authentication for HP-UX

Perform the following steps to configure SSH Public Key Authentication on HP-UX:

1. Uncomment the following lines in the /etc/ssh/sshd_config file:

   ```
   PermitRootLogin yes
   PubkeyAuthentication yes
   AuthorizedKeysFile .ssh/authorized_keys
   ```

   > **Note:**
   >
   > Change the value of `PermitRootLogin` to `yes` only if it does not violate local security policies. If Public Key Authentication is enabled, then you can change the value of `PermitRootLogin` to `without-password`.
   >
   > Instead of using the root account, if you can use a user account with sudo privileges, then you do not need to perform this step.

2. To restart the SSH Server, enter the following command:

   ```
   /opt/ssh/sbin/sshd
   ```

3. To test login, enter the following command:

   ```
   ssh -i /.ssh/id_rsa -l root server_IP_address
   ```

   When prompted, enter the passkey to connect to the server.

   > **Note:**
   >
   > Instead of using the root account, you can use a user account with sudo privileges, if desired.

4. Set the `privateKey[LOADFROMURL]` advanced settings parameter to include the complete path of the `id_rsa` file with the prefix `file://`.

   For example:

   ```
   file:///OIM_HOME/server/ConnectorDefaultDirectory/SSH/config/id_rsa
   ```

2.2.2.7.3 Configuring SSH Public Key Authentication for Linux

Perform the following steps to configure SSH Public Key Authentication on Linux:

1. Enter the following commands to restart the UNIX server:

   ```
   /etc/init.d/sshd stop
   /etc/init.d/sshd start
   ```

2. Copy the /.ssh/id_rsa file to the following directory:

   ```
   OIM_HOME/server/ConnectorDefaultDirectory/SSH/config
   ```

3. To check if you can connect to the target system using the SSH protocol, directly from the command prompt and without using a password, enter the following command:

   > **Note:**
   >
   > The account used to run the OIM application server on UNIX should have the ownership of the id_rsa file.

   ```
   ssh -i OIM_HOME/server/ConnectorDefaultDirectory/SSH/config/id_rsa -l root
   host_ip_address
   ```

4. Set the `privateKey[LOADFROMURL]` advanced settings parameter to include the complete path of the `id_rsa` file with the prefix `file://`.

   For example:

   ```
   file:///OIM_HOME/server/ConnectorDefaultDirectory/SSH/config/id_rsa
   ```

2.2.2.7.4 Configuring SSH Public Key Authentication for AIX

Perform the following steps to configure SSH Public Key Authentication on AIX:

1. Use the /etc/ssh/sshd_config file to set the following parameters:

```
export PATH=$PATH: /usr/sbin
Installation path: /etc/ssh/
sshd -- /usr/sbin/
```

2. Open the /etc/ssh/sshd_config file, and uncomment the following lines:

   ```
   AuthorizedKeysFile .ssh/authorized_keys
   PermitRootLogin yes
   PubkeyAuthentication yes
   ```

   > **Note:**
   >
   > Change the value of `PermitRootLogin` to `yes` only if it does not violate local security policies. If Public Key Authentication is enabled, then you can change the value of `PermitRootLogin` to `without-password`.
   >
   > Instead of using the root account, if you can use a user account with sudo privileges, then you do not need to perform this step.

3. To restart the SSH server, enter the following command:

   ```
   /usr/sbin/sshd
   ```

4. To test the login, enter the following command:

   ```
   ssh -i /.ssh/id_rsa -l root server_IP_address
   ```

   When prompted, enter the passkey to connect to the server.

   > **Note:**
   >
   > Instead of using the root account, you can use a user account with sudo privileges, if desired.

5. Set the `privateKey[LOADFROMURL]` advanced settings parameter to include the complete path of the `id_rsa` file with the prefix `file://`.

   For example:

   ```
   file:///OIM_HOME/server/ConnectorDefaultDirectory/SSH/config/id_rsa
   ```

2.3 Creating an Application By Using the Connector

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

The following is the high-level procedure to create an application by using the connector:

> **Note:**
>
> For detailed information on each of the steps in this procedure, see Creating Applications of *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

1. Create an application in Identity Self Service. The high-level steps are as follows:
   1. Log in to Identity Self Service either by using the **System Administration** account or an account with the **ApplicationInstanceAdministrator** admin role.
   2. Ensure that the **Connector Package** option is selected when creating an application.
   3. Update the basic configuration parameters to include connectivity-related information.
   4. If required, update the advanced setting parameters to update configuration entries related to connector operations.
   5. Review the default user account attribute mappings. If required, add new attributes or you can edit or delete existing attributes.
   6. Review the provisioning, reconciliation, organization, and catalog settings for your application and customize them if required. For example, you can customize the default correlation rules for your application if required.
   7. Review the details of the application and click **Finish** to submit the application details.

      The application is created in Oracle Identity Governance.
   8. When you are prompted whether you want to create a default request form, click **Yes** or **No**.

      If you click **Yes**, then the default form is automatically created and is attached with the newly created application. The default form is created with the same name as the application. The default form cannot be modified later. Therefore, if you want to customize it, click **No** to manually create a new form and attach it with your application.
2. Verify reconciliation and provisioning operations on the newly created application.

> **See Also:**
>
> - Configuring the UNIX Connector for details on basic configuration and advanced settings parameters, default user account attribute mappings, default correlation rules, and reconciliation jobs that are predefined for this connector
> - Configuring Oracle Identity Governance for details on creating a new form and associating it with your application, if you chose not to create the default form

3 Configuring the UNIX Connector

While creating a target or an authoritative application, you must configure connection-related parameters that the connector uses to connect Oracle Identity Governance with your target system and perform connector operations. In addition, you can view and edit attribute mappings between the process form fields in Oracle Identity Governance and target system columns, predefined correlation rules, situations and responses, and reconciliation jobs.

3.1 Basic Configuration Parameters

These are the connection-related parameters that Oracle Identity Governance requires to connect to UNIX. These parameters are common for both target applications and authoritative applications.

Table 3-1 Basic Configuration Parameters for UNIX

Parameter	Mandatory?	Description
host	Yes	Host name or the IP address of the target system computer
loginUser	Yes	User ID of the administrator to perform connector operations `root` or `jdoe` Here, `jdoe` can be the SUDO user ID, for the SUDO Admin mode. Alternatively, on Solaris, it can be the user ID of the account to which you assign the minimum privileges required to perform connector operations. See Creating a Target System SUDO User Account for Connector Operations for more information.
loginUserpassword	Yes	Password of the administrator

Table 3-1 (Cont.) Basic Configuration Parameters for UNIX

Parameter	Mandatory?	Description
loginShellPrompt	No	Shell prompt that you encounter when you login to the target system using the loginUser account Default value: `[#$]` **Note:** This value is a regular expression. By default, the connector works if the shell prompt on the target system is either # or $. However, if the shell prompt is different, for example >, then you must change the value of this parameter to the actual prompt. To know the loginShellPrompt, perform the following steps on the target system: 1. Log in to the target system using the user and the password specified in the loginUser and loginUserPassword parameters. Note the login prompt. For example, `#.` 2. Run the `sh` command. Note the shell prompt, if it is different from the previous prompt. For example, `$.` 3. Run the `sudo -k` command. Note the shell prompt, if it is different from the previous prompt. For example, `$.` 4. Run the `sudo -v` command. This will prompt you for the password if loginUser is a SUDO user. Enter the password and continue. Note the shell prompt, if it is different from the previous prompt. For example, `$.` 5. Run the `sudo -s` command. Note the shell prompt, if it is different from the previous prompt. For example, `$.` For the values shown in the examples, the loginShellPrompt parameter value should be `[#$].` In addition, if the shell prompt displayed in any of the previous steps is similar to `home/jdoe>,` then the prompt is > (not the entire string, `home/jdoe>`).
port	No	Port at which the SSH or Telnet service is running on the server Default value for SSH: 22 Default value for Telnet: 23
Connector Server Name	No	Name of the IT resource of type "Connector Server". By default, this field is blank. If you use a Connector Server, then a default IT resource is created during application creation whose default value is: `UNIX Connector Server`

Table 3-1 (Cont.) Basic Configuration Parameters for UNIX

Parameter	Mandatory?	Description
connectionType	No	Protocol used by the connector to connect to the target system The connector supports the following connection types: • `SSH` - Used for SSH with password-based authentication. • `SSHPUBKEY` - Used for SSH with key-based authentication. • `TELNET` - Used for Telnet connection. Default value: `SSH`
connectorPrompt	No	Shell prompt set by the connector for its operations on the target system Default value: `#@#` **Note:** If this value occurs in user login names, comment fields, directory names, and so on, some connector operations may be affected. In such a case, the value for the connector prompt can be changed to a value that does not occur in the names.
passphrase	No	Passphrase for the key file to use with key based authentication **Note:** You must provide a passphrase if you use key-based authentication.

Table 3-1 (Cont.) Basic Configuration Parameters for UNIX

Parameter	Mandatory?	Description
propertyFileName	No	Relative path of the ScriptProperties.properties file of the target system You can leave this field blank if you want to use the default scripts. However, if you want to use custom scripts other than the OOTB scripts, then you must provide a value for this field. The connector will try to determine the path of the properties file by running the `uname -a` command on the target system. If the connector is unable to determine an appropriate value (when an exception is encountered), then it will display the following error message: `Unable to determine UNIX Type. Please provide property file name in IT Resource.` In the case of an error message, enter one of the following values (or a different path if you want to use customized scripts) depending on the target system and the user account: • `scripts/solaris/sudo/ScriptProperties.properties` • `scripts/solaris/nonsudo/ScriptProperties.properties` • `scripts/linux/sudo/ScriptProperties.properties` • `scripts/linux/nonsudo/ScriptProperties.properties` • `scripts/aix/sudo/ScriptProperties.properties` • `scripts/aix/nonsudo/ScriptProperties.properties` • `scripts/hpux/sudo/ScriptProperties.properties` • `scripts/hpux/nonsudo/ScriptProperties.properties`
rbacAuthorization	No	Indicates whether the user provided in the loginUser parameter is a RBAC user Default value: `false` See Creating an RBAC User Account for Connector Operations on Solaris for more information.
rbacRoleName	No	If you specify the rbacAuthorization parameter as `true`, then enter the name of the role assigned to the RBAC user. Otherwise, do not specify a value for this parameter.
rbacRolePassword	No	If you specify the rbacAuthorization parameter as `true`, then enter the password of the role assigned to the RBAC user. Otherwise, do not specify a value for this parameter.
sudoAuthorization	No	Indicates whether the user provided in the loginUser parameter is a SUDO user Default value: `false`

3.2 Advanced Settings Parameters

These are the configuration-related entries that the connector uses during reconciliation and provisioning operations. Unless specified, the parameters in the table are applicable to both target and authoritative applications.

Table 3-2 Advanced Settings Parameters for UNIX

Parameter	Mandatory?	Description
Connector Name	Yes	This parameter holds the name of the connector class. Default value: `org.identityconnectors.genericunix.GenericUnixConnector`
defaultConnectorShelll	No	This is the defaultShell used for connector operations. Do not modify this entry unless you are using RBAC. Default value: `sh` **Note:** If you are using RBAC, then the decode value must be changed from `sh` to `pfsh.`
Bundle Name	Yes	Name of the connector bundle package. Default value: `org.identityconnectors.genericunix` Do not modify this entry.
Bundle Version	Yes	Version of the connector bundle class. Do not modify this entry. Default value: `12.3.0`
targetDateFormat	No	Format of the date on the target system as: MM/dd/yy **Note:** You must ensure to enter the correct Java date format for the target system. An incorrect format may affect provisioning of the Expire Date attribute. For information about the date format, see `http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html` and `http://docs.oracle.com/javase/6/docs/api/java/text/DateFormat.html`.
whitelistRegex	No	Specifies characters that are allowed as a part of the field values. For example: The regular expression, `[A-Za-z0-9_//]*`, allows all alphanumeric, underscore, and forward slash characters. You can add more characters if needed. **Note:** For information about the supported regular expressions, you can refer to a guide such as `http://www.zytrax.com/tech/web/regex.htm` This regular expression does not apply to the GECOS field, which can have any characters.

Table 3-2 (Cont.) Advanced Settings Parameters for UNIX

Parameter	Mandatory?	Description
sudoPasswdExpectExpression	No	Regular expression for the password prompt displayed on the target system when you enter the SUDO mode. If the target system displays a different prompt, then you must change this password prompt. Default value: `password:` **Note:** The third-party library, Expect4j, matches these expected expressions to the actual contents of the console output on the UNIX target system. Therefore, you must ensure that these fields have correct values. Incorrect values may impact the connector operations.
rbacRoleExpectExpressions	No	Regular expressions for the two comma-separated prompts. Default value: `password:,[$#]` The first prompt (password:) is the password prompt displayed on the Solaris target system when you enter the SUDO mode for the RBAC role. If the target system displays a different prompt, then you must change this password prompt. The second prompt ([$#]) is the shell prompt displayed after running the previous command in SUDO mode. If the target system displays a different prompt, then you must change this shell prompt. **Note:** The third-party library, Expect4j, matches these expected expressions to the actual contents of the console output on the UNIX target system. Therefore, you must ensure that these fields have correct values. Incorrect values may impact the connector operations.
commandTimeout	No	Time in milliseconds for which the connector would wait for a response from the target system. After this time, the connector will throw a timeout exception. **Default value:** `10000000` You can increase this value if you encounter a 'command timed out' exception for connector operations.
configPropertiesOnScripts	No	Lists the properties that are sent to the scripts: `moveHomeDirContents,shadow,defaultHomeBaseDir,defaultPriGroup,defaultShell,nisPwdDir,nisBuildDirectory,removeHomeDirContents,forceDeleteUserHome,syncToken,mirrorFilesLocation,connectorPrompt` For example, if during provisioning, you want to set a default shell for the users. To do so: 1. Verify that the 'defaultShell' property is a part of this list. 2. Add an entry for this property. Set the value for defaultShell to `/bin/sh.` If the target-specific script supports the defaultShell property, it would be set. Not all scripts support all the attributes listed. You must manually check the script contents for supported attributes.

Table 3-2 (Cont.) Advanced Settings Parameters for UNIX

Parameter	Mandatory?	Description

mirrorFilesLocation	No	Directory used by the connector to store copies of the /etc/passwd and shadow files:
		• For a Target application:
		`/etc/connector_mirror_files`
		• For an Authoritative application:
		`/etc/connector_mirror_files_trusted`
		Note: This directory has to be manually created on the target before performing reconciliation. If you want to specify a different directory, ensure that the directory exists on the target system and the Login User has read-write access to the directory.

passwordExpectExpressions	No	Regular expression for the two comma-separated password prompts that are displayed on the target system when a password is set for a user: `new[\s](unix[\s])?password:,new[\s](unix[\s])?password([\s]again)?:`
		Note: The third-party library, Expect4j, matches these expected expressions to the actual contents of the console output on the UNIX target system.
		Therefore, you must ensure that these fields have correct values. Incorrect values may impact the connector operations.
		If the regular expression does not work on your target system, then you can specify the exact prompts in this entry.
		For example, if you set the password for a user and you get the following prompt:
		`Enter Password for USER1:`
		`Re-enter Password for USER1:`
		Then, you can set the value as follows:
		`enter password,re-enter password`

supportedLanguage	No	Shell script language supported on the target system **Default value**: Bourne
telnetAuthenticationPrompts **Note:** This entry is applicable for Telnet connection, when the Connection Type parameter is set to TELNET.	No	The login and password prompts on a target system using Telnet connection. Default value: `login:,Password:` **Note:** The third-party library, Expect4j, matches these expected expressions to the actual contents of the console output on the UNIX target system. Therefore, you must ensure that these fields have correct values. Incorrect values may impact the connector operations.
moveHomeDirContents	No	Specifies whether the old home directory contents should be moved to the new directory location when changing the Home Directory. Default value: `true`

Table 3-2 (Cont.) Advanced Settings Parameters for UNIX

Parameter	Mandatory?	Description
privateKey	No	Path to the id_rsa file. Sample value: `file:///scratch/files/jars/unix/id_rsa_linux`
Pool Max Idle	No	Maximum number of idle objects in a pool. Sample value: 10
Pool Max Size	No	Maximum number of connections that the pool can create. Sample value: 10
Pool Max Wait	No	Maximum time, in milliseconds, the pool must wait for a free object to make itself available to be consumed for an operation. Sample value: 150000
Pool Min Evict Idle Time	No	Minimum time, in milliseconds, the connector must wait before evicting an idle object. Sample value: 120000
Pool Min Idle	No	Minimum number of idle objects in a pool. Sample value: 1

3.3 Attribute Mappings

The attribute mappings on the Schema page vary depending on whether you are creating a target application or an authoritative application.

- Attribute Mappings for a Target Application
- Attribute Mappings for an Authoritative Application

3.3.1 Attribute Mappings for a Target Application

The Schema page for a Target application displays the default schema (provided by the connector) that maps Oracle Identity Governance attributes to target system columns. The connector uses these mappings during reconciliation and provisioning operations.

UNIX User Account Attributes

This table lists the mapping of attribute between the process form fields and UNIX columns. The table also lists whether a specific attribute is used during provisioning or reconciliation and whether it is a matching key field for fetching records during reconciliation.

If required, you can edit the default attribute mappings by adding new attributes or deleting existing attributes as described in Creating a Target Application of *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Note:

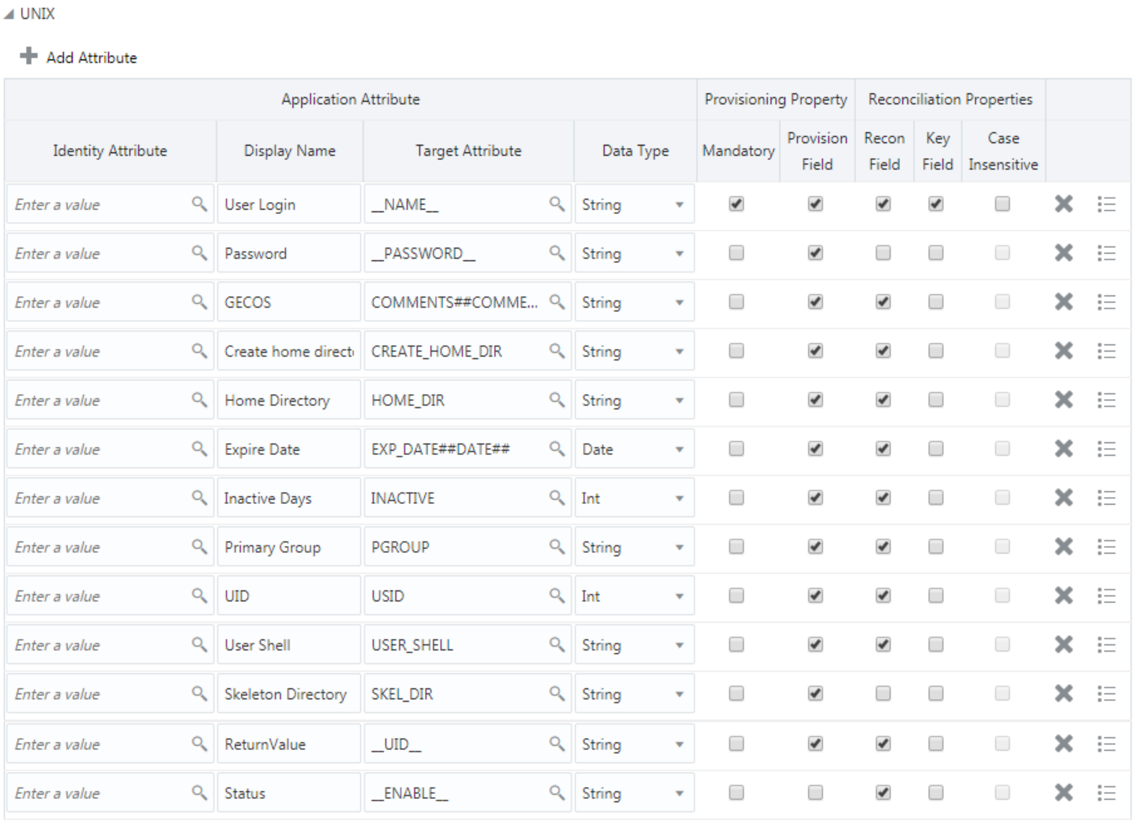
Whenever you edit the default attribute mapping, you must make corresponding updates to the attributes in the scripts pertaining to your target system. If you add or update an attribute for provisioning, then update the provisioning scripts as described in Updating the Scripts for Provisioning After Editing Schema Attributes. If you add or update an attribute for reconciliation, then update the reconciliation scripts as described in Updating the Scripts for Reconciliation After Editing Schema Attributes.

Table 3-3 Default Attribute Mappings for UNIX User Account

Display Name	Target Attribute	Data Type	Mandatory Provisioning Property?	Provision Field?	Recon Field	Key Field?	Case Insensitive?
User Login	__NAME__	String	Yes	Yes	Yes	Yes	Not Applicable
GECOS	COMMENTS##COMMENTS##	String	No	Yes	Yes	No	Not Applicable
Create home directory	CREATE_HOME_DIR	String	No	Yes	Yes	No	Not Applicable
Home Directory	HOME_DIR	String	No	Yes	Yes	No	Not Applicable
Expire Date	EXP_DATE##DATE##	Date	No	Yes	Yes	No	Not Applicable
Inactive Days	INACTIVE	Int	No	Yes	Yes	No	Not Applicable
Primary Group	PGROUP	String	No	Yes	Yes	No	Not Applicable
UID	USID	Int	No	Yes	Yes	No	Not Applicable
User Shell	USER_SHELL	String	No	Yes	Yes	No	Not Applicable
ReturnValue	__UID__	String	No	Yes	Yes	No	Not Applicable
Status	__ENABLE__	String	No	No	Yes	No	Not Applicable
Password	__PASSWORD__	String	No	Yes	No	No	Not Applicable
Skeleton Directory	SKEL_DIR	String	No	Yes	No	No	Not Applicable

Figure 3-1 shows the default User account attribute mapping.

Figure 3-1 Default Attribute Mappings for UNIX User Account



◢ UNIX

✚ Add Attribute

Application Attribute				Provisioning Property		Reconciliation Properties				
Identity Attribute	Display Name	Target Attribute	Data Type	Mandatory	Provision Field	Recon Field	Key Field	Case Insensitive		
Enter a value	User Login	__NAME__	String	☑	☑	☑	☑	□	✖	☰
Enter a value	Password	__PASSWORD__	String	□	☑	□	□	□	✖	☰
Enter a value	GECOS	COMMENTS##COMME...	String	□	☑	☑	□	□	✖	☰
Enter a value	Create home directo	CREATE_HOME_DIR	String	□	☑	☑	□	□	✖	☰
Enter a value	Home Directory	HOME_DIR	String	□	☑	☑	□	□	✖	☰
Enter a value	Expire Date	EXP_DATE##DATE##	Date	□	☑	☑	□	□	✖	☰
Enter a value	Inactive Days	INACTIVE	Int	□	☑	☑	□	□	✖	☰
Enter a value	Primary Group	PGROUP	String	□	☑	☑	□	□	✖	☰
Enter a value	UID	USID	Int	□	☑	☑	□	□	✖	☰
Enter a value	User Shell	USER_SHELL	String	□	☑	☑	□	□	✖	☰
Enter a value	Skeleton Directory	SKEL_DIR	String	□	☑	□	□	□	✖	☰
Enter a value	ReturnValue	__UID__	String	□	☑	☑	□	□	✖	☰
Enter a value	Status	__ENABLE__	String	□	□	☑	□	□	✖	☰

Secondary Group Entitlement Attributes

This is the default mapping of attributes between process form fields and secondary group list-related columns in the target system. The table lists whether a given attribute is mandatory during provisioning. It also lists whether a given attribute is used during reconciliation and whether it is a matching key field for fetching records during reconciliation.

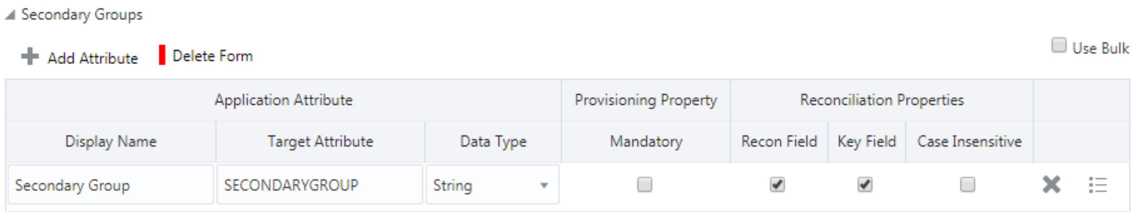
If required, you can edit the default attribute mappings by adding new attributes or deleting existing attributes as described in Creating a Target Application of *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Table 3-4 Default Attribute Mappings for Secondary Group Entitlement

Display Name	Application Attribute	Data Type	Mandatory Provisioning Property?	Recon Field	Key Field?	Case Insensitive ?
Secondary Group	SECONDARYGROUP	String	No	Yes	Yes	No

Figure 3-2 shows the default Secondary Group entitlement mapping.

Figure 3-2 Default Attribute Mappings for Secondary Group Entitlement



Application Attribute			Provisioning Property	Reconciliation Properties			
Display Name	Target Attribute	Data Type	Mandatory	Recon Field	Key Field	Case Insensitive	
Secondary Group	SECONDARYGROUP	String	☐	☑	☑	☐	✖ ☰

3.3.2 Attribute Mappings for an Authoritative Application

The Schema page for an Authoritative application displays the default schema (provided by the connector) that maps Oracle Identity Governance attributes to target system columns. The connector uses these mappings during reconciliation and provisioning operations.

Generic UNIX Trusted User Account Attributes

Table 3-5 lists the mapping of attributes between the reconciliation fields in Oracle Identity Governance and UNIX attributes. The table also lists the data type for a given attribute and specifies whether it is a mandatory attribute for reconciliation.

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

You may use the default schema that has been set for you or update and change it before continuing to the next step.

> **Note:**
>
> Whenever you edit the default attribute mapping, you must make corresponding updates to the attributes in the scripts pertaining to your target system. If you add or update an attribute for reconciliation, then update the reconciliation scripts as described in Updating the Scripts for Reconciliation After Editing Schema Attributes.

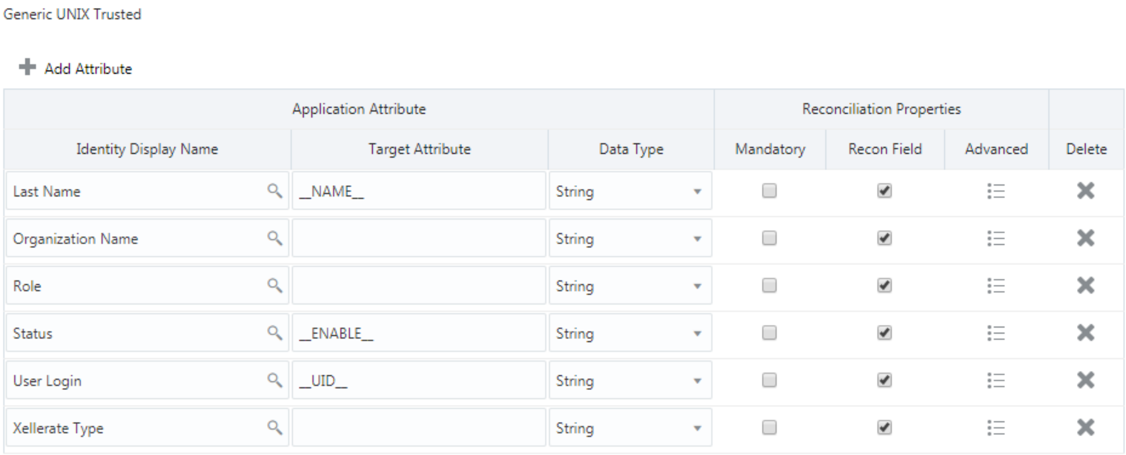
The Organization Name, Role, and Xellerate Type identity attributes are mandatory fields on the OIG User form. They cannot be left blank during reconciliation. The target attribute mappings for these identity attributes are empty by default because there are no corresponding columns in the target system. Therefore, the connector provides default values (as listed in the “Default Value for Identity Display Name” column of Table 3-5) that it can use during reconciliation. For example, the default target attribute value for the Organization Name attribute is Xellerate Users. This implies that the connector reconciles all target system user accounts into the Xellerate Users organization in Oracle Identity Governance. Similarly, the default attribute value for Xellerate Type attribute is End-User, which implies that all reconciled user records are marked as end users.

Table 3-5 Default Attribute Mappings for Generic UNIX Trusted User Account

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

Figure 3-3 shows the default User account attribute mapping.

Figure 3-3 Default Attribute Mappings for Generic UNIX Trusted User Account



Generic UNIX Trusted

Add Attribute

Application Attribute			Reconciliation Properties			
Identity Display Name	Target Attribute	Data Type	Mandatory	Recon Field	Advanced	Delete
Last Name	_NAME_	String	☐	☑	☰	✖
Organization Name		String	☐	☑	☰	✖
Role		String	☐	☑	☰	✖
Status	_ENABLE_	String	☐	☑	☰	✖
User Login	_UID_	String	☐	☑	☰	✖
Xellerate Type		String	☐	☑	☰	✖

3.4 Rules, Situations, and Responses

Learn about the predefined rules, responses and situations for target and authoritative applications. The connector use these rules and responses for performing reconciliation.

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

3.4.1 Rules, Situations, and Responses for a Target Application

Learn about the predefined rules, responses and situations for a Target application. The connector use these rules and responses for performing reconciliation.

Predefined Identity Correlation Rules

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

Table 3-6 lists the default simple correlation rule for the UNIX connector. If required, you can edit the default correlation rule or add new rules. You can create complex correlation rules also. For more information about adding or editing simple or complex correlation rules, see Updating Identity Correlation Rule in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Table 3-6 Predefined Identity Correlation Rule for a UNIX Target Application

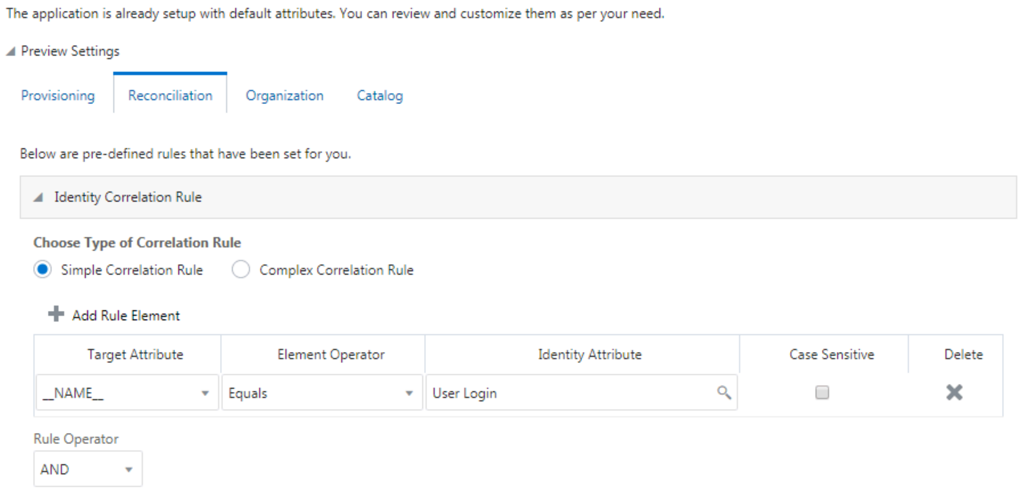
Target Attribute	Element Operator	Identity Attribute	Case Sensitive?
__NAME__	Equals	User Login	No

In this identity rule:

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

Figure 3-4 shows the simple correlation rule for the UNIX connector.

Figure 3-4 Simple Correlation Rule for a UNIX Target Application



The application is already setup with default attributes. You can review and customize them as per your need.

◢ Preview Settings

Below are pre-defined rules that have been set for you.

◢ Identity Correlation Rule

Choose Type of Correlation Rule

● Simple Correlation Rule ○ Complex Correlation Rule

Target Attribute	Element Operator	Identity Attribute	Case Sensitive	Delete
NAME	Equals	User Login	□	✖

Rule Operator

AND

Predefined Situations and Responses

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

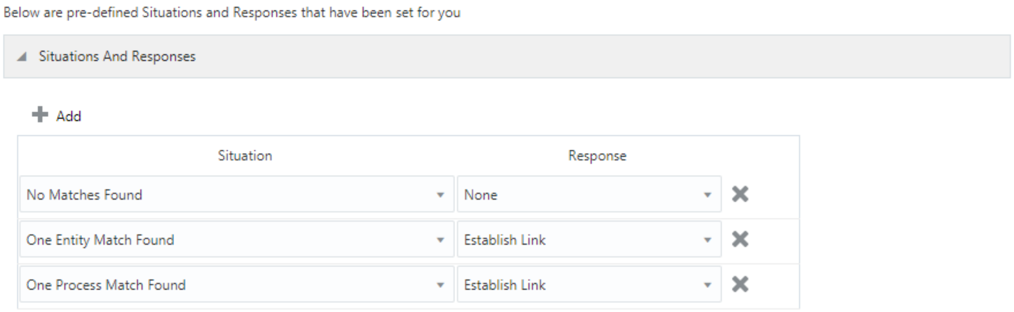
Table 3-7 lists the default situations and responses for the UNIX connector. If required, you can edit these default situations and responses or add new ones. For more information about adding or editing situations and responses, see Creating a Target Application in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Table 3-7 Predefined Situations and Responses for a UNIX Target Application

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

Figure 3-5 shows the situations and responses that the connector provides by default.

Figure 3-5 Predefined Situations and Responses for a UNIX Target Application



Below are pre-defined Situations and Responses that have been set for you

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

3.4.2 Rules, Situations, and Responses for an Authoritative Application

Learn about the predefined rules, responses and situations for an Authoritative application. The connector use these rules and responses for performing reconciliation.

Predefined Identity Correlation Rules

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

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

Table 3-8 lists the default simple correlation rule for the UNIX connector. If required, you can edit the default correlation rule or add new rules. You can create complex correlation rules also. For more information about adding or editing simple or complex correlation rules, see Updating Identity Correlation Rule in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Table 3-8 Predefined Identity Correlation Rule for a UNIX Authoritative Application

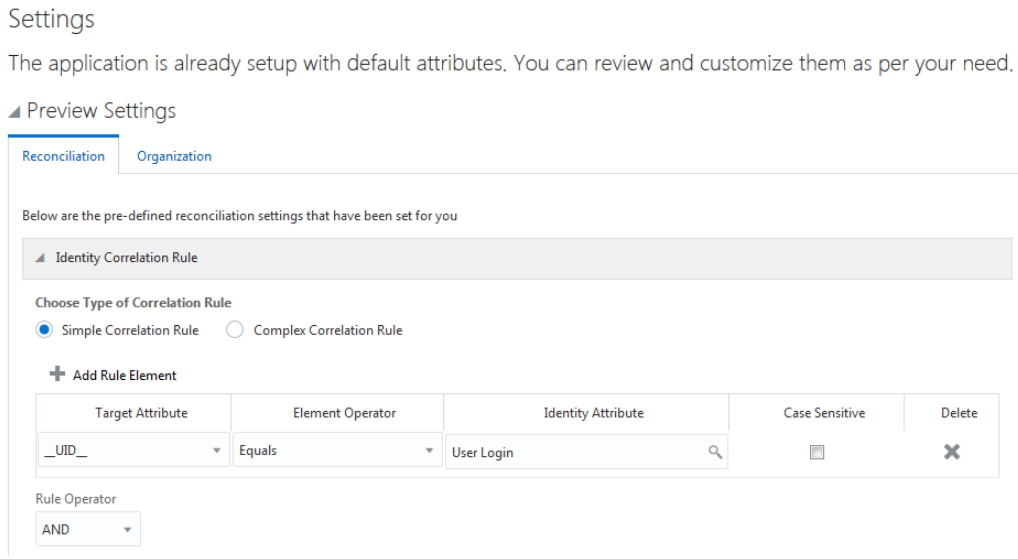
Target Attribute	Element Operator	Identity Attribute	Case Sensitive?
__UID__	Equals	User Login	No

In this identity rule:

- __UID__ is an attribute on the target system that uniquely identifies the user account.
- User Login is the field on the OIG User form.

Figure 3-6 shows the simple correlation rule for the UNIX connector.

Figure 3-6 Simple Correlation Rule for a UNIX Authoritative Application



Settings

The application is already setup with default attributes. You can review and customize them as per your need.

Preview Settings

Below are the pre-defined reconciliation settings that have been set for you

Identity Correlation Rule

Choose Type of Correlation Rule

● Simple Correlation Rule ○ Complex Correlation Rule

Target Attribute	Element Operator	Identity Attribute	Case Sensitive	Delete
__UID__	Equals	User Login	□	✖

Rule Operator: AND

Predefined Situations and Responses

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

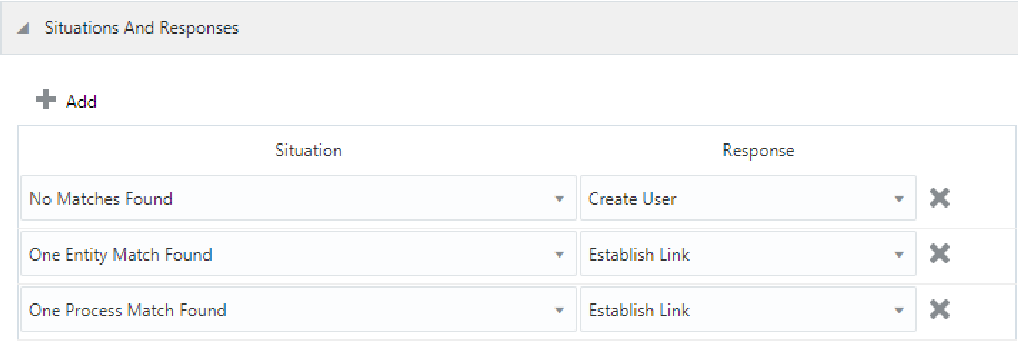
Table 3-9 lists the default situations and responses for the UNIX connector. If required, you can edit these default situations and responses or add new ones. For more information about adding or editing situations and responses, see Creating an Authoritative Application in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Table 3-9 Predefined Situations and Responses for a UNIX Authoritative Application

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

Figure 3-7 shows the situations and responses that the connector provides by default.

Figure 3-7 Predefined Situations and Responses for a UNIX Authoritative Application



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

3.5 Reconciliation Jobs

Learn about reconciliation jobs that are automatically created in Oracle Identity Governance after you create a target or an authoritative application for your target system.

- Reconciliation Jobs for a Target Application
- Reconciliation Jobs for an Authoritative Application

3.5.1 Reconciliation Jobs for a Target Application

These are the reconciliation jobs that are automatically created in Oracle Identity Governance after you create a Target application for your target system.

You can either use these predefined jobs or edit them to meet your requirements. Alternatively, you can create custom reconciliation jobs. For information about editing these predefined jobs or creating new ones, see Updating Reconciliation Jobs in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Full User Reconciliation Job

The UNIX Target Resource Full User Reconciliation job is used to fetch all user records from the target system.

Table 3-10 Parameters of the UNIX Target Resource Full User Reconciliation Job

Parameter	Value
Application Name	Name of the application you created for your target system. This value is the same as the value that you provided for the Application Name field while creating your target application. Do not modify this value.
Batch Size	Specify the number of records that must be included in each batch Default value: `0` See Performing Batched Reconciliation for more information.
Batch start index	Specify the position from which the records will be included in each batch Default value: `0`
Filter	Enter the expression for filtering records that the scheduled job must reconcile. Sample value: `equalTo('__UID__','SEPT12USER1')` For information about the filters expressions that you can create and use, see ICF Filter Syntax in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*.
No. of Batches	Specify the total number of batches that must be reconciled. Default value: `0`
Object Type	Type of object you want to reconcile. Default value: `User`

Incremental User Reconciliation Job

The UNIX Target Incremental Resource User Reconciliation job is used to fetch the records that are added or modified after the last reconciliation run.

Table 3-11 Parameters of the UNIX Target Incremental Resource User Reconciliation Job

Parameter	Value
Application Name	Name of the application you created for your target system. This value is the same as the value that you provided for the Application Name field while creating your target application. Do not modify this value.
Batch Size	Specify the number of records that must be included in each batch Default value: `0` See Performing Batched Reconciliation for more information.

Table 3-11 (Cont.) Parameters of the UNIX Target Incremental Resource User Reconciliation Job

Parameter	Value
Batch start index	Specify the position from which the records will be included in each batch Default value: `0`
No. of Batches	Specify the total number of batches that must be reconciled. Default value: `0`
Object Type	Type of object you want to reconcile Default value: `User`
Scheduled Task Name	Name of the scheduled task **Note**: For the scheduled task shipped with this connector, you must not change the value of this attribute. However, if you create a copy of the task, then you can enter the unique name for that scheduled task as the value of this attribute.
Sync Token	Time stamp at which the last reconciliation run started **Note**: Do not enter a value for this attribute. The reconciliation engine automatically enters a value for this attribute. If you set this attribute to an empty value, then incremental reconciliation operations fetch all the records (perform full reconciliation).

Reconciliation Jobs for Entitlements

The following jobs are available for reconciling entitlements:

- UNIX User Primary Group Lookup Reconciliation
- UNIX User Shell Lookup Reconciliation

The parameters for all the reconciliation jobs are the same.

Table 3-12 Parameters of the Reconciliation Jobs for Entitlements

Parameter	Description
Application Name	Name of the application you created for your target system. This value is the same as the value that you provided for the Application Name field while creating your target application. Do not modify this value.
Code Key Attribute	Enter the name of the connector or target system attribute that is used to populate the Code Key column of the lookup definition (specified as the value of the Lookup Name attribute). Default value: `__NAME__` **Note:** Do not change the value of this attribute.

Table 3-12 (Cont.) Parameters of the Reconciliation Jobs for Entitlements

Parameter	Description
Decode Attribute	Enter the name of the connector or target system attribute that is used to populate the Decode column of the lookup definition (specified as the value of the Lookup Name attribute). Default value: `__NAME__`
Lookup Name	This parameter holds the name of the lookup definition that maps each lookup definition with the data source from which values must be fetched. Depending on the reconciliation job you are using, the default values are as follows: • For UNIX User Primary Group Lookup Reconciliation: `Lookup.UNIX.PrimaryGroup` • For UNIX User Shell Lookup Reconciliation: `Lookup.UNIX.UserShell`
Object Type	Enter the type of object whose values must be synchronized. Depending on the scheduled job you are using, the default values are as follows: • For UNIX User Primary Group Lookup Reconciliation: `Group` • For UNIX User Shell Lookup Reconciliation: `__SHELLS__` **Note:** Do not change the value of this attribute.

3.5.2 Reconciliation Jobs for an Authoritative Application

These are the reconciliation jobs that are automatically created in Oracle Identity Governance after you create an Authoritative application for your target system.

You can either use these predefined jobs or edit them to meet your requirements. Alternatively, you can create custom reconciliation jobs. For information about editing these predefined jobs or creating new ones, see Updating Reconciliation Jobs in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Full User Reconciliation Job

The UNIX User Trusted Recon job is used to fetch all user records from the target system.

Table 3-13 Parameters of the UNIX User Trusted Recon Job

Parameter	Value
Application Name	Name of the application you created for your target system. This value is the same as the value that you provided for the Application Name field while creating your target application. Do not modify this value.

Table 3-13 (Cont.) Parameters of the UNIX User Trusted Recon Job

Parameter	Value
Batch Size	Specify the number of records that must be included in each batch Default value: `0` See Performing Batched Reconciliation for more information.
Batch start index	Specify the position from which the records will be included in each batch Default value: `0`
Filter	Enter the expression for filtering records that the scheduled job must reconcile. Sample value: `equalTo('__UID__','SEPT12USER1')` For information about the filters expressions that you can create and use, see ICF Filter Syntax in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*.
No. of Batches	Specify the total number of batches that must be reconciled. Default value: `0`
Object Type	Type of object you want to reconcile. Default value: `User`

Incremental User Reconciliation Job

The UNIX User Trusted Incremental Recon job is used to fetch the records that are added or modified after the last reconciliation run.

Table 3-14 Parameters of the UNIX User Trusted Incremental Recon Job

Parameter	Value
Application Name	Name of the application you created for your target system. This value is the same as the value that you provided for the Application Name field while creating your target application. Do not modify this value.
Batch Size	Specify the number of records that must be included in each batch Default value: `0` See Performing Batched Reconciliation for more information.
Batch start index	Specify the position from which the records will be included in each batch Default value: `0`
No. of Batches	Specify the total number of batches that must be reconciled. Default value: `0`

Table 3-14 (Cont.) Parameters of the UNIX User Trusted Incremental Recon Job

Parameter	Value
Object Type	Type of object you want to reconcile Default value: `User`
Scheduled Task Name	Name of the scheduled task **Note**: For the scheduled task shipped with this connector, you must not change the value of this attribute. However, if you create a copy of the task, then you can enter the unique name for that scheduled task as the value of this attribute.
Sync Token	Time stamp at which the last reconciliation run started **Note**: Do not enter a value for this attribute. The reconciliation engine automatically enters a value for this attribute. If you set this attribute to an empty value, then incremental reconciliation operations fetch all the records (perform full reconciliation).

4 Performing the Postconfiguration Tasks for the UNIX Connector

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

- Configuring Oracle Identity Governance
- Harvesting Entitlements and Sync Catalog
- Managing Logging
- Configuring the IT Resource for the Connector Server
- Localizing Field Labels in UI Forms

4.1 Configuring Oracle Identity Governance

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

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

The following topics describe the procedures to configure Oracle Identity Governance:

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

4.1.1 Creating and Activating a Sandbox

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

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

4.1.2 Creating a New UI Form

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

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

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

4.1.3 Publishing a Sandbox

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

1. In Identity System Administration, deactivate the sandbox.
2. Log out of Identity System Administration.
3. Log in to Identity Self Service using the xelsysadm user credentials and then activate the sandbox that you deactivated in Step 1.
4. In the Catalog, ensure that the application instance form for your resource appears with correct fields.
5. Publish the sandbox. See Publishing a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*.

4.1.4 Updating an Existing Application Instance with a New Form

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

To update an existing application instance with a new form:

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

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

4.2 Harvesting Entitlements and Sync Catalog

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

To harvest entitlements and sync catalog:

1. Run the scheduled jobs for lookup field synchronization listed in Reconciliation Jobs.
2. Run the Entitlement List scheduled job to populate Entitlement Assignment schema from child process form table.
3. Run the Catalog Synchronization Job scheduled job.

> **See Also:**
>
> Predefined Scheduled Tasks in *Oracle Fusion Middleware Administering Oracle Identity Governance* for a description of the Entitlement List and Catalog Synchronization Job scheduled jobs

4.3 Managing Logging

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

The following topics provide detailed information about logging:

- Understanding Log Levels
- Enabling Logging

4.3.1 Understanding Log Levels

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

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

- SEVERE.intValue()+100

  This level enables logging of information about fatal errors.
- SEVERE

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

  This level enables logging of information about potentially harmful situations.
- INFO

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

- CONFIG

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

- FINE, FINER, FINEST

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

These message types are mapped to ODL message type and level combinations as shown in Table 4-2.

Table 4-1 Log Levels and ODL Message Type:Level Combinations

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

Table 4-2 Log Levels and ODL Message Type:Level Combinations

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

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

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

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

4.3.2 Enabling Logging

Perform this procedure to enable logging in Oracle WebLogic Server.

To enable logging:

1. Edit the logging.xml file as follows:

   a. Add the following blocks in the file:

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

   b. Replace both occurrences of **`[LOG_LEVEL]`** with the ODL message type and level combination that you require. Understanding Log Levels 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='unix-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.GENERICUNIX" level="NOTIFICATION:1"
      useParentHandlers="false">
           <handler name="telnetssh-handler"/>
           <handler name="console-handler"/>
         </logger>
      ```

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

2. Save and close the file.

3. Set the following environment variable to redirect the server logs to a file:

   For Microsoft Windows:

   ```
   set WLS_REDIRECT_LOG=FILENAME
   ```

   For UNIX:

```
export WLS_REDIRECT_LOG=FILENAME
```

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

4. Restart the application server.

4.4 Configuring the IT Resource for the Connector Server

If you have used the Connector Server, then you must configure values for the parameters of the Connector Server IT resource.

After you create the application for your target system, the connector creates a default IT resource for the Connector Server. The name of this default IT resource is `UNIX Connector Server`.

In Oracle Identity System Administration, search for and edit the UNIX Connector Server IT resource to specify values for the parameters of IT resource for the Connector Server listed in Table 4-3. For more information about searching for IT resources and updating its parameters, see Managing IT Resources in *Oracle Fusion Middleware Administering Oracle Identity Governance*.

Table 4-3   Parameters of the IT Resource for the UNIX Connector Server

Parameter	Description
Host	Enter the host name or IP address of the computer hosting the Connector Server. Sample value: `HostName`
Key	Enter the key for the Connector Server.
Port	Enter the number of the port at which the Connector Server is listening. By default, this value is blank. You must enter the port number that is displayed on the terminal when you start the Connector Server. For example: `8763`
Timeout	Enter an integer value which specifies the number of milliseconds after which the connection between the Connector Server and Oracle Identity Governance times out. If the value is zero or if no value is specified, the timeout is unlimited. Recommended value: `0`
UseSSL	Enter `true` to specify that you will configure SSL between Oracle Identity Governance and the Connector Server. Otherwise, enter `false.` Default value: `false` **Note:** It is recommended that you configure SSL to secure communication with the connector server. To configure SSL, see Configuring SSL for Java Connector Server in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*.

4.5 Localizing Field Labels in UI Forms

You can localize UI form field labels by using the resource bundle corresponding to the language you want to use. The resource bundles are available in the connector installation package.

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

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

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

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

6. Edit the BizEditorBundle.xlf file in the following manner:
   a. Search for the following text:

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

   b. Replace with the following text:

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

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

      ```
      <file source-language="en" target-language="fr"
      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 UNIX 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_UNIX_GRPNAME__c_description']}">
      <source>Primary Group</source>
      </target>
      </trans-unit>
      <trans-unit
      id="sessiondef.oracle.iam.ui.runtime.form.model.UNIX.entity.UNIXEO.UD_UNI
      X_GRPNAME__c_LABEL">
      <source>Primary Group</source>
      </target>
      </trans-unit>
      ```

d. Open the resource file from the connector package, for example UNIX_fr.properties, and get the value of the attribute from the file, for example, global.udf.UD_UNIX_GRPNAME= Groupe principal.

e. Replace the original code shown in Step 6.c with the following:

```
<trans-unit id="$
{adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceBundl
e']
['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.userEO.
UD_UNIX_GRPNAME__c_description']}">
<source> Primary Group</source>
<target> Groupe principal</target>
</trans-unit>
<trans-unit
id="sessiondef.oracle.iam.ui.runtime.form.model.UNIX.entity.UNIXEO.UD_UNI
X_GRPNAME__c_LABEL">
<source> Primary Group</source>
<target> Groupe principal</target>
</trans-unit>
```

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

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

   Sample file name: BizEditorBundle_fr.xlf.

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

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

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

5 Using the UNIX Connector

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

- Configuring Reconciliation
- Configuring Reconciliation Jobs
- Performing Provisioning Operations
- Uninstalling the Connector

5.1 Configuring Reconciliation

Reconciliation involves duplicating in Oracle Identity Governance the creation of and modifications to user accounts on the target system.

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

- Performing Full Reconciliation
- Performing Limited Reconciliation
- Performing Batched Reconciliation
- Performing Incremental Reconciliation

5.1.1 Performing Full Reconciliation

Full reconciliation involves reconciling all existing user records from the target system into Oracle Identity Governance. After you create the application, you must first perform full reconciliation.

To perform a full reconciliation run, remove (delete) any value currently assigned to the Filter attribute of the Full User Reconciliation job. See Reconciliation Jobs for information about this reconciliation job.

During a full reconciliation run, if you provide both batching parameters and filters, the connector processes the data in batches. Then, filters are applied to the processed data.

5.1.2 Performing Limited Reconciliation

You can perform limited reconciliation by creating filters for the reconciliation module, and reconcile records from the target system based on a specified filter criterion.

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

This connector provides a Filter attribute (a scheduled task attribute) that allows you to use any of the UNIX resource attributes to filter the target system records.

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

While creating the application, follow the instructions in Configuring Reconciliation Jobs to specify attribute values.

5.1.3 Performing Batched Reconciliation

You can perform batched reconciliation to reconcile a specific number of records from the target system into Oracle Identity Governance.

During a reconciliation run, all changes in the target system records are reconciled into Oracle Identity Governance. Depending on the number of records to be reconciled, this process may require a large amount of time. In addition, if the connection breaks during reconciliation, then the process would take longer to complete.

You can configure batched reconciliation to avoid these problems.

To configure batched reconciliation, you must specify values for the following user reconciliation scheduled task attributes:

- Batch Size: Use this attribute to specify the number of records that must be included in each batch.
- Batch Start Index: Use this attribute to specify the position from which the records will be included in each batch.
- No. of Batches: Use this attribute to specify the total number of batches that must be reconciled.

By default, the values of all attributes is `0,` indicating that all records will be included (no batched reconciliation). The following example illustrates this:

Suppose that of a total 314 records, only 200 records were processed before encountering an exception or an error. During the next reconciliation run, you can set Batch Start Index to 200 to process the records from 200 to 314.

You specify values for these attributes by following the instructions described in Configuring Reconciliation Jobs.

5.1.4 Performing Incremental Reconciliation

During incremental reconciliation, only records that are added or modified after the last reconciliation run are fetched into Oracle Identity Governance.

You can perform incremental recon by running the UNIX Target Incremental Resource User Reconciliation or UNIX User Trusted Incremental Recon jobs described in Reconciliation Jobs.

The following is the behavior of incremental reconciliation scheduled tasks:

- Incremental reconciliation scheduled tasks do not support filtering of records.
- Incremental reconciliation scheduled tasks fetch data from the target system in alphabetical order.

- If you run an incremental reconciliation scheduled task for the first time, or if you run the task after removing the value of **Sync Token** parameter, then the following directories (or the directory specified in the configuration lookup definition) must be empty:

  connector_mirror_files

  connector_mirror_files_trusted
- After an incremental reconciliation scheduled task completes, the following files will be generated in the **connector_mirror_files** or **connector_mirror_files_trusted** directory (or in the directory specified in the configuration lookup definition). Here, SYNC_TOKEN refers to the value of the Sync Token parameter.
  - *SYNC_TOKEN.***passwd** file contains previous copy of the password file in the /etc directory, for example, `/etc/passwd`.
  - *SYNC_TOKEN.***shadow** file contains previous copy of the shadow file in the /etc directory, for example, `/etc/shadow`.
  - *SYNC_TOKEN.***group** file contains previous copy of the group file in the /etc directory, for example, `/etc/group`.
  - **passwd_difference_incr** file contains differences between the `/etc/passwd` and the *SYNC_TOKEN.*`passwd` files.
  - **shadow_difference_incr** file contains differences between the `/etc/shadow` and *SYNC_TOKEN.*`shadow` files.
  - **group_difference_incr** file contains differences between the `/etc/group` and *SYNC_TOKEN.*`group` files.
  - **record** file contains the actual records that will be sent back to Oracle Identity Governance in alphabetically sorted order.

5.2 Configuring Reconciliation Jobs

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

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

To configure a reconciliation job:

1. Log in to Identity System Administration.
2. In the left pane, under System Management, click **Scheduler**.
3. Search for and open the scheduled job as follows:
   a. In the Search field, enter the name of the scheduled job as the search criterion. Alternatively, you can click **Advanced Search** and specify the search criterion.
   b. In the search results table on the left pane, click the scheduled job in the Job Name column.
4. On the Job Details tab, you can modify the parameters of the scheduled task:

- **Retries**: Enter an integer value in this field. This number represents the number of times the scheduler tries to start the job before assigning the Stopped status to the job.
- **Schedule Type**: Depending on the frequency at which you want the job to run, select the appropriate schedule type. See Creating Jobs in *Oracle Fusion Middleware Administering Oracle Identity Governance*.

In addition to modifying the job details, you can enable or disable a job.

5. On the **Job Details** tab, in the Parameters region, specify values for the attributes of the scheduled task.

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

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

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

5.3 Performing Provisioning Operations

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

To perform provisioning operations in Oracle Identity Governance:

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

See Also:

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

5.4 Uninstalling the Connector

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

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

For example: `UNIX User; UNIX Group`

Note:

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

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

6

Extending the Functionality of the UNIX Connector

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

6.1 Configuring the Connector for a New Target System

You can configure the connector to support an additional flavor of UNIX.

> **Note:**
>
> Perform this procedure only if you want to configure the connector for an additional flavor of UNIX other than the target systems listed in Certified Components.

By default, the connector uses pre-configured scripts to support Linux, Solaris, AIX, and HP-UX.

The scripts are available in the connector bundle JAR file. You can download the bundle from Oracle Identity Governance database using the DownloadJars utility in *OIM_HOME/* bin directory or from the installation media. If you are using Connector Server, then you can copy the bundle from *CONNECTOR_SERVER_HOME/* bundles directory.

You can add similar scripts with similar directory structure to support an additional flavor of UNIX. For example, you can add connector support for a target system with BSD/OS flavor of UNIX. To do so:

1. Create the following directories, which will be packaged into the connector bundle JAR:

   scripts/bsdos/nonsudo/

   scripts/bsdos/sudo/

2. Create the following scripts for sudo and non-sudo authentication types. Then, drop them in the corresponding directories created in the previous step.

Note:

It is recommended that the script files have read-only permissions.

Table 6-1 Custom Scripts to Support New Flavor of UNIX Target System

Script Name	Description
CreateNativeUser.txt	Create a user on target
DeleteNativeUser.txt	Delete a user from target
FetchAllGroupRecords.txt	For group lookup reconciliation
FetchAllUserRecords.txt	For full user reconciliation
FetchAllShellRecords.txt	For Shell lookup reconciliation
FetchSingleUserRecord.txt	Get one user
NativeUserIncrementalRecon.txt	Used by SyncOp for incremental reconciliation
UpdateNativeUser.txt	For user updates

3. Create and update the ScriptProperties.properties file with details of all the scripts.

   The values should be paths to the new scripts. See the scripts/linux/ScriptProperties.properties file for sample values. For example:

   ```
   CREATE_USER_SCRIPT=scripts/bsdos/sudo/CreateNativeUser.txt
   DELETE_USER_SCRIPT=scripts/bsdos/sudo/DeleteNativeUser.txt
   FETCH_SINGLE_USER=scripts/bsdos/sudo/FetchSingleUserRecord.txt
   FETCH_FULL_RECON_SCRIPT=scripts/bsdos/sudo/FetchAllUserRecords.txt
   INCREMENTAL_RECON_SCRIPT=scripts/bsdos/sudo/NativeUserIncrementalRecon.txt
   ```

4. Ensure that the values returned by the scripts are appropriate format, as expected by the bundle. See scripts/linux/ for sample scripts.

5. Create and update the ResponseMapping.properties file in the scripts/bsdos directory.

   The ResponseMapping.properties file contains mapping between the message to be expected and the exception class with which the message has to be wrapped and thrown. See the scripts/linux/ResponseMapping.properties file for sample values. For example:

   ```
   User already
   exists=org.identityconnectors.framework.common.exceptions.AlreadyExistsExcept
   ion
   Group already
   exists=org.identityconnectors.framework.common.exceptions.AlreadyExistsExcept
   ion
   ```

6. Run the following command to update the bundle JAR file with the new scripts:

   ```
   jar uvf org.identityconnectors.genericunix-1.0.0.jar scripts/bsdos/
   ```

7. In the PropertyFileName basic configuration parameter, specify the value of the path to the properties file.

   For example: `scripts/bsdos/nonsudo/ScriptProperties.properties` (for non-sudo authentication)

See Basic Configuration Parameters for more information about the PropertyFileName parameter.

6.2 Configuring the Connector for Multiple Installations of the Target System

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

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

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

6.3 Configuring Transformation and Validation of Data

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

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

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

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

6.4 Configuring Action Scripts

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

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

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

6.5 Updating the Scripts for Reconciliation After Editing Schema Attributes

The connector provides a default set of attribute mappings that are displayed on the Schema page as described in Attribute Mappings.

If you add to or edit the default attribute mappings for reconciliation, you must make corresponding updates to the attributes in the reconciliation scripts.

To update the reconciliation scripts:

1. Copy or download the connector bundle JAR file.

   You can download the bundle from Oracle Identity Governance database using the DownloadJars utility in *OIM_HOME*/bin directory or from the installation media. If you are using Connector Server, then you can copy the bundle from *CONNECTOR_SERVER_HOME*/bundles directory.

2. Extract the JAR file to edit the scripts.

   For example, to extract the script for Linux, non-sudo user for reconciliation, run the following command:

   ```
   jar xvf org.identityconnectors.genericunix-12.3.0.jar scripts/linux/nonsudo/
   FetchAllUserRecords.txt
   ```

   The FetchAllUserRecords.txt script is called when you run full reconciliation without the equalsTo filter. If you want, you can edit additional reconciliation scripts:

   - FetchSingleUserRecord.txt - this script is called when you run full reconciliation with the equalsTo filter.
   - NativeUserIncrementalRecon.txt - this script is called during incremental reconciliation.

3. Open the script for editing.

   > **Note:**
   >
   > You must have a good knowledge of bash scripts to edit the scripts. Before editing the scripts, you can create a backup of the original scripts.

   For example, if you have added the __GID__ attribute for reconciliation, you can construct a block for the __GID__ attribute similar to other blocks.

   Add the following line after line 9 starting with PGROUP to fetch the __GID__ field:

   ```
   __GID__=$( id -G $__NAME__ | cut -d' ' -f1);
   ```

   Add an entry to line 32 starting with RESULT as follows:

   ```
   RESULT=__NAME__:$__NAME__:__GID__:$__GID__:__ENABLE__:$ENABLE
   ```

   Add an entry to line 41 starting with unset as follows:

```
unset inputline __NAME__ USID COMMENTS HOME_DIR USER_SHELL PGROUP secgrplist
__GID__;
```

> **See Also:**
>
> Sample Scripts for Updating Default Attributes for Reconciliation for the original and updated FetchAllUserRecords.txt script

4. Save the script and update the bundle as follows:

   ```
   jar uvf org.identityconnectors.genericunix-12.3.0.jar scripts/linux/nonsudo/
   FetchAllUserRecords.txt
   ```

5. Replace the old bundle by using UpdateJars utility in *OIM_HOME*/bin directory.

   If you are using the Connector Server, stop it. Then, replace the JAR in the *CONNECTOR_SERVER_HOME*/bundles directory and restart the Connector Server.

6.6 Updating the Scripts for Provisioning After Editing Schema Attributes

The connector provides a default set of attribute mappings that are displayed on the Schema page as described in Attribute Mappings.

If you add to or edit the default attribute mappings for provisioning, you must make corresponding updates to the attributes in the provisioning scripts.

To update the provisioning scripts:

1. Copy or download the connector bundle JAR file.

   You can download the bundle from Oracle Identity Governance database using the DownloadJars utility in *OIM_HOME*/bin directory or from the installation media. If you are using Connector Server, then you can copy the bundle from *CONNECTOR_SERVER_HOME*/bundles directory.

2. Extract the JAR file to edit the scripts.

   For example, to extract the script for Linux, non-sudo user for provisioning, run the following command:

   ```
   jar xvf org.identityconnectors.genericunix-12.3.0.jar scripts/linux/nonsudo/
   CreateNativeUser.txt
   ```

   This script is used to enable create operations on the newly added attribute. Similarly, you can edit the UpdateNativeUser.txt script to enable update operations.

3. Open the script for editing.

   > **Note:**
   >
   > You must have a good knowledge of bash scripts to edit the scripts. Before editing the scripts, you can create a backup of the original scripts.

For example, if you have added the __GID__ attribute for provisioning, you can construct a block for the __GID__ attribute similar to other blocks, as follows (lines 76 to 78):

```
    if [ ! -z $__GID__ ] ;then
        command="$command -g $__GID__";
    fi;
```

Add an entry to line 91 starting with unset as follows:

```
unset defaultHomeBaseDir homedir checkHomeBaseDir grp defaultPriGroup
__GID__;
```

> **See Also:**
>
> Sample Scripts for Updating Default Attributes for Provisioning for the original and updated CreateNativeUser.txt script

4. Save the script and update the bundle as follows:

   ```
   jar uvf org.identityconnectors.genericunix-12.3.0.jar scripts/linux/nonsudo/
   CreateNativeUser.txt
   ```

5. Replace the old bundle by using UpdateJars utility in *OIM_HOME/*bin directory.

   If you are using the Connector Server, stop it. Then, replace the JAR in the *CONNECTOR_SERVER_HOME/*bundles directory and restart the Connector Server.

7 Upgrading the UNIX Connector

If you have already deployed the 11.1.1.7.0 version of the UNIX connector, then you can upgrade the connector to version 12.2.1.3.0 by uploading the new connector JAR files to the Oracle Identity Manager database.

> **Note:**
>
> - If you have deployed the 11.1.1.6.0 or earlier version of the UNIX connector, you must first upgrade the connector to version 11.1.1.7.0. See Upgrading the Connector in *Oracle Identity Manager Connector Guide for UNIX*.
> - Before you perform the upgrade procedure:
>   - It is strongly recommended that you create a backup of the Oracle Identity Manager database. Refer to the database documentation for information about creating a backup.
>   - As a best practice, perform the upgrade procedure in a test environment initially.

The following sections discuss the procedure to upgrade the connector:

- Preupgrade Steps
- Upgrade Steps
- Postupgrade Steps

> **See Also:**
>
> Upgrading Connectors in *Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed information on these steps

7.1 Preupgrade Steps

Preupgrade steps involve performing a reconciliation run and defining the source connector.

Perform the following preupgrade steps:

1. Perform a reconciliation run to fetch all latest updates to Oracle Identity Manager.
2. Define the source connector (an earlier release of the connector that must be upgraded) in Oracle Identity Manager. You define the source connector to update

the Deployment Manager XML file with customization changes made to the connector.

3. If required, create the connector XML file for a clone of the source connector.

7.2 Upgrade Steps

This is a summary of the procedure to upgrade the connector for both staging and production environments.

Depending on the environment in which you are upgrading the connector, perform one of the following steps:

- Staging Environment

  Perform the upgrade procedure by using the wizard mode.

  > **Note:**
  >
  > Do not upgrade IT resource type definition. In order to retain the default setting, you must map the IT resource definition to "None".

- Production Environment

  Perform the upgrade procedure by using the silent mode.

See Managing Connector Lifecycle in *Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed information about the wizard and silent modes.

7.3 Postupgrade Steps

Postupgrade steps involve uploading new connector jars, configuring the upgraded IT resource of the source connector, deploying the Connector Server, and configuring the latest token value of the scheduled job.

Perform the following procedure:

1. Upload new connector JARs as follows:
   a. Run the Upload JARs utility ($*ORACLE_HOME*/bin/UploadJars.sh) for uploading connector JARs.
   b. Upload bundle/org.identityconnectors.genericunix-12.3.0.jar as ICFBundle.

      > **Note:**
      >
      > If you need to add a third-party JAR:
      >
      > - Navigate to the bundle directory.
      > - Create /lib folder and drop the third party jar in that folder.
      > - Update the bundle with library "jar uvf org.identityconnectors.genericunix-12.3.0.jar lib/*FILE_NAME*".

   c. Upload lib/GenericUnix-oim-integration.jar as JavaTask.

2. Replicate all changes made to the Form Designer of the Design Console in a new UI form as follows:

   a. Log in to Oracle Identity System Administration.

   b. Create and activate a sandbox.

   c. Create a new UI form to view the upgraded fields.

   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 2.c) and then save the application instance.

   e. Publish the sandbox.

3. Configure the upgraded IT resource of the source connector.

4. Deploy the Connector Server.

5. Configure the sync token value for the following scheduled jobs:

   - UNIX Target Incremental Resource User Reconciliation
   - UNIX User Trusted Incremental Recon

   After upgrading the connector, you can perform either full reconciliation or incremental reconciliation. This ensures that records created or modified since the last reconciliation run are fetched into Oracle Identity Manager. From the next reconciliation run onward, the reconciliation engine automatically enters a value for the Sync Token attribute.

   > **See Also:**
   >
   > - Configuring Oracle Identity Governance for information about creating, activating, or publishing a sandbox and creating a new UI form
   > - Using an Identity Connector Server in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about deploying the Connector Server
   > - Configuring Reconciliation for more information about performing full or incremental reconciliation

8

Testing and Troubleshooting the UNIX Connector

After you create the application, you must test the connector to ensure that it functions as expected.

This chapter provides details on the following topics related to connector testing:

- Testing the UNIX Connector
- Troubleshooting the UNIX Connector

8.1 Testing the UNIX Connector

You can use the testing utility, supplied with the connector package, to identify the cause of problems associated with connecting to the target system and performing basic operations on the target system.

The test-utility directory of connector installation media contains the following files:

- The example-config.groovy file is a sample configuration that can be used to set the connection properties of the target system and the connector.
- The README file contains instructions to configure and run the testing utility.
- The test-utility.jar file contains the class files used by the testing utility.

> **Note:**
>
> The testing utility does not support separate update and delete actions. You can update and delete a user while creating the user in the same operation.

To use the testing utility, perform the following steps:

1. Ensure JDK 1.6 is installed.
2. Unzip the connector zip file.
3. Locate and switch to the `test-utility` directory in the contents of the extracted zip file.

   The test-utility.jar and example-config.groovy files already exist in this directory.

> **Note:**
>
> On a Microsoft Windows computer, the bundle package (`org.identityconnectors.genericunix-1.0.0.jar`) must be in a directory other than the test-utility directory.

4. Copy the following JAR files to the test-utility directory:
   - `connector-framework.jar`
   - `connector-framework-internal.jar`
   - `groovy-all.jar`

   > **Note:**
   >
   > These are files are delivered as part of the OIM EAR application, and they are located in the `oim.ear/APP-INF/lib` directory.
   >
   > You must add these three JAR files to the `JAVA_HOME/jre/lib/ext` directory to run the test utility.
   >
   > If you copy these JAR files into the test-utility directory, you can exclude these files in the Java command provided in Step 6.

5. Update the `example-config.groovy` file to reflect your local settings.
6. Run one of the following commands:

   On a UNIX computer:

   ```
   java -classpath ./test-utility.jar:./connector-framework.jar:./
   connector-framework-internal.jar:./groovy-all.jar:./test-utility.jar
   oracle.iam.connectors.testutility.Main example-config.groovy | tee test.log
   ```

   On a Microsoft Windows computer (assuming the current directory is c:\test-utility):

   ```
   java -classpath C:\test-utility\test-utility.jar;C:\test-utility\connector-
   framework.jar;C:\test-utility\connector-framework-internal.jar;C:\test-
   utility\groovy-all.jar;C:\test-utility\test-utility.jar;
   oracle.iam.connectors.testutility.Main example-config.groovy
   ```

8.2 Troubleshooting the UNIX Connector

You can troubleshoot the UNIX connector depending on the type of error you encounter.

The following sections list solutions to some commonly encountered errors of the following types:

- Connection Errors
- Create User Errors
- Delete User Errors
- Edit User Errors

- TimeOut Errors

8.2.1 Connection Errors

This table lists the solution to a commonly encountered Connection error.

Problem Description	Solution
Oracle Identity Governance cannot establish a connection to the target system. **Returned Error Message:** UNIX connection exception	• Ensure that the target system is running. • Ensure that Oracle Identity Governance is working (that is, the database is running). • Ensure that all the adapters have been compiled. • Ensure that values for all the basic configuration parameters have been correctly specified.

8.2.2 Create User Errors

This table lists the solution to a commonly encountered Create User error.

Problem Description	Solution
Oracle Identity Governance cannot create a user. **Returned Error Message:** User already exists	A user with the assigned ID already exists in the target system.

8.2.3 Delete User Errors

This table lists the solution to a commonly encountered Delete User error.

Problem Description	Solution
Oracle Identity Governance cannot delete a user. **Returned Error Message:** User does not exist in target system	The specified user does not exist in the target system.

8.2.4 Edit User Errors

This table lists the solution to a commonly encountered Edit User error.

Problem Description	Solution
Oracle Identity Governance cannot update a user. **Returned Error Message:** User does not exist in target system	Review the log for more details.

8.2.5 TimeOut Errors

This table lists the solution to a commonly encountered TimeOut error.

Problem Description	Solution
Connection timeout during connector operations, typically during reconciliation operations. **Returned Error Message:** Command timed out	Increase the value of the Command Timeout entry. See Advanced Settings Parameters for more information.
Reconciliation fails with an error. **Returned Error Message:** org.identityconnectors.genericunix.common.GenericUnixUtil :getMatchedResponseInfo : Buffer:-> > > > >Password:	Change the value of the Sudo Passwd Expect Expression entry from Password: to `password.` **Note**: The letter "P" must be in lowercase. See Advanced Settings Parameters for more information.

9

Known Issues

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

- User Remains in the Provisioned State After a Trusted Source Deleted Reconciliation Run
- Incorrect Response Message Appears During a Provisioning Operation
- Parent Form Is Not Linked to the Child Form After Upgrade
- Other Encrypted Parameters Appear to Be Updated After Updating an Encrypted Parameter

9.1 User Remains in the Provisioned State After a Trusted Source Deleted Reconciliation Run

After performing a trusted delete reconciliation run, a user remains in the Provisioned state if the user is deleted from Oracle Identity Governance.

This issue occurs whenever identity data is reconciled authoritatively from a target system to Oracle Identity Governance and daily account updates are provisioned from Oracle Identity Governance back to the target system.

Workaround:

Run the target delete reconciliation scheduled task after running the trusted delete reconciliation task. After the account reconciliation is complete, the scheduled task converts the account status to the Revoked state.

9.2 Incorrect Response Message Appears During a Provisioning Operation

During a provisioning operation, if the connector encounters a timeout error, an incorrect response message is displayed in the resource history of the scheduled task.

```
Response: CONNECTOR_EXCEPTION
```

```
Response Description: Unknown response received
```

Workaround:

There is no workaround available for this issue.

9.3 Parent Form Is Not Linked to the Child Form After Upgrade

After upgrading the connector, the parent form is not linked to the child form.

Workaround:

You must create a new version of the parent form and make it active.

9.4 Other Encrypted Parameters Appear to Be Updated After Updating an Encrypted Parameter

If you update an encrypted basic configuration parameter such as loginUserPassword, other encrypted parameters such as passphrase and rbacRolePassword also appear to have been updated.

Workaround:

There is no workaround available for this issue.

A

Privileges Required for Performing Provisioning and Reconciliation

These are the privileges required for successful provisioning operations and reconciliation runs.

This appendix includes the following topics:

- Privileges Required for Running Commands on Solaris and Linux
- Privileges Required for Running Commands on HP-UX
- Privileges Required for Running Commands on AIX

A.1 Privileges Required for Running Commands on Solaris and Linux

These are the privileges and permissions required to run commands on Solaris and Linux for performing provisioning operations and reconciliation runs.

Users must have privileges to run the following commands:

```
usermod, useradd, userdel, passwd, chage, id, cut, touch, awk, uniq
```

In addition, the users must have execute permissions for the following commands:

```
sed, cat, diff, sort, rm, grep, egrep, echo, /usr/bin/sh, /bin/sh
```

Users must have read and write permissions on the `/home`, `/tmp` and `/etc/connector_mirror_files` (or the mirror files directory specified in the configuration lookup definition) directories.

A.2 Privileges Required for Running Commands on HP-UX

These are the privileges and permissions required to run commands on HP-UX for performing provisioning operations and reconciliation runs.

Users must have privileges to run the following commands:

```
modprpw, usermod, useradd, userdel, passwd, chage, id, cut, touch, awk,
uniq
```

In addition, the users must have execute permissions for the following commands:

```
sed, cat, diff, sort, rm, grep, egrep, echo, /usr/bin/sh, /bin/sh
```

Users must have read and write permissions on the `/home`, `/tmp` and `/etc/connector_mirror_files` (or the mirror files directory specified in the configuration lookup definition) directories.

A.3 Privileges Required for Running Commands on AIX

These are the privileges and permissions required to run commands on AIX for performing provisioning operations and reconciliation runs.

User must have privileges to execute the following commands:

`mkuser, useradd, chuser, rmuser, lsuser, /usr/bin/usermod, /usr/chuser`, `id, cut, touch, awk, uniq`

In addition, the users must have execute permissions for the following commands:

`/usr/bin/bdiff, sh, cat, /usr/bin/sort, /usr/bin/rm, /usr/bin/grep, /bin/echo, /bin/sed, command.`

Users must have read and write permissions on the `/home`, `/tmp` and `/etc/connector_mirror_files` (or the mirror files directory specified in the configuration lookup definition) directories.

B

Sample Scripts for Updating Default Attributes for Reconciliation

This appendix provides sample scripts for editing the default attribute mappings for reconciliation.

Sample scripts for the procedure described in Updating the Scripts for Reconciliation After Editing Schema Attributes are as follows:

- Original Sample Script
- Updated Sample Script

B.1 Original Sample Script

This is the original FetchAllUserRecords.txt script.

```
while read inputline ;do
    __NAME__=$(echo $inputline | cut -d: -f1);
    USID=$(echo $inputline | cut -d: -f3);
    COMMENTS=$(echo $inputline | cut -d: -f5);
    HOME_DIR=$(echo $inputline | cut -d: -f6);
    CREATE_HOME_DIR="false";
    if [ -d "$HOME_DIR" ] ;then CREATE_HOME_DIR="true";fi;
    USER_SHELL=$(echo $inputline | cut -d: -f7);
    PGROUP=$( id -G -n $__NAME__ | cut -d' ' -f1);
    shadowRecord=$(cat /etc/shadow |grep $__NAME__);
    INACTIVE=$(echo $shadowRecord | cut -d: -f7);
    EXP_DATE=$(echo $shadowRecord | cut -d: -f8);
    secgrplist="";
    id -G -n $__NAME__ | grep -q " ";
    if [ $? -eq 0 ];then
        secgrplist=$( id -G -n $__NAME__ | cut -d ' ' -f2- | sed 's/ /~~~/g');
    fi;
    ENABLE="true";
    if [ ! -z "$__NAME__" ] ;then
        passwd -S $__NAME__ | grep -w LK >/dev/null;
         if [ $? -eq 0 ] ;then
            ENABLE="false";
         fi;
         passwd -S $__NAME__ | grep -w locked >/dev/null;
         if [ $? -eq 0 ] ;then
            ENABLE="false";
         fi;
    fi;

RESULT1=__NAME__:$__NAME__:__ENABLE__:$ENABLE:COMMENTS:$COMMENTS:USID:$USID:USER_
SHELL:$USER_SHELL:HOME_DIR:$HOME_DIR:;
        RESULT2=CREATE_HOME_DIR:$CREATE_HOME_DIR:SECONDARYGROUP:$secgrplist:;

RESULT3=PGROUP:$PGROUP:INACTIVE:$INACTIVE:EXP_DATE:$EXP_DATE:__UID__:$__NAME__;
        RESULT="$RESULT1$RESULT2$RESULT3";
    echo "$RESULT" | grep -q -w "$connectorPrompt";
```

```
    if [ $? -gt 0 ];then
        echo "RESULT_START $RESULT RESULT_END";
    else
        echo "Record contains connector prompt. Hence ignored";
    fi;
done < /etc/passwd;[ $? -eq 0 ] && echo "SUCCESS";
unset inputline __NAME__ USID COMMENTS HOME_DIR USER_SHELL PGROUP secgrplist;
unset ENABLE passwordFull passwordF passwordS RESULT RESULT1 RESULT2 RESULT3
__UID__ INACTIVE EXP_DATE shadowRecord;
```

B.2 Updated Sample Script

This is the FetchAllUserRecords.txt script that has been updated to include the newly added __GID__ attribute. The updated lines are represented in bold font.

```
while read inputline ;do
    __NAME__=$(echo $inputline | cut -d: -f1);
    USID=$(echo $inputline | cut -d: -f3);
    COMMENTS=$(echo $inputline | cut -d: -f5);
    HOME_DIR=$(echo $inputline | cut -d: -f6);
    CREATE_HOME_DIR="false";
    if [ -d "$HOME_DIR" ] ;then CREATE_HOME_DIR="true";fi;
    USER_SHELL=$(echo $inputline | cut -d: -f7);
    PGROUP=$( id -G -n $__NAME__ | cut -d' ' -f1);
    __GID__=$( id -G $__NAME__ | cut -d' ' -f1);
    shadowRecord=$(cat /etc/shadow |grep $__NAME__);
    INACTIVE=$(echo $shadowRecord | cut -d: -f7);
    EXP_DATE=$(echo $shadowRecord | cut -d: -f8);
    secgrplist="";
    id -G -n $__NAME__ | grep -q " ";
    if [ $? -eq 0 ];then
        secgrplist=$( id -G -n $__NAME__ | cut -d ' ' -f2- | sed 's/ /~~~/g');
    fi;
    ENABLE="true";
    if [ ! -z "$__NAME__" ] ;then
        passwd -S $__NAME__ | grep -w LK >/dev/null;
         if [ $? -eq 0 ] ;then
            ENABLE="false";
         fi;
         passwd -S $__NAME__ | grep -w locked >/dev/null;
         if [ $? -eq 0 ] ;then
            ENABLE="false";
         fi;
    fi;

RESULT1=__NAME__:$__NAME__:__ENABLE__:$ENABLE:COMMENTS:$COMMENTS:USID:$USID:USER_
SHELL:$USER_SHELL:HOME_DIR:$HOME_DIR:;
        RESULT2=CREATE_HOME_DIR:$CREATE_HOME_DIR:SECONDARYGROUP:$secgrplist:;

RESULT3=PGROUP:$PGROUP:INACTIVE:$INACTIVE:EXP_DATE:$EXP_DATE:__UID__:$__NAME__:__
GID__:$__GID__;
        RESULT="$RESULT1$RESULT2$RESULT3";
    echo "$RESULT" | grep -q -w "$connectorPrompt";
    if [ $? -gt 0 ];then
        echo "RESULT_START $RESULT RESULT_END";
    else
        echo "Record contains connector prompt. Hence ignored";
    fi;
done < /etc/passwd;[ $? -eq 0 ] && echo "SUCCESS";
unset inputline __NAME__ USID COMMENTS HOME_DIR USER_SHELL PGROUP secgrplist
```

```
__GID__;
unset ENABLE passwordFull passwordF passwordS RESULT RESULT1 RESULT2 RESULT3
__UID__ INACTIVE EXP_DATE shadowRecord;
```

C

Sample Scripts for Updating Default Attributes for Provisioning

This appendix provides sample scripts for editing the default attribute mappings for provisioning.

Sample scripts for the procedure described in Updating the Scripts for Provisioning After Editing Schema Attributes are as follows:

- Original Sample Script
- Updated Sample Script

C.1 Original Sample Script

This is the original CreateNativeUser.txt script.

```
if [ ! -z "$__UID__" ] ;then
    __NAME__=$__UID__;
else
    __NAME__=$__NAME__;
fi;
if id $__NAME__ > /dev/null 2>&1 ;then
    echo "User already exists";
else
    globalVar="true";
    if [ ! -z $SECONDARYGROUP ] ;then
        command="$command -G $SECONDARYGROUP";
    fi;

    homedir="";
    if [ ! -z $HOME_DIR ] ;then
        homedir=$HOME_DIR;
    else
        if [ ! -z $defaultHomeBaseDir ] ;then
            homedir=$defaultHomeBaseDir;
        fi;
    fi;
    if [ ! -z $homedir ] ;then
        checkHomeBaseDir=$(test -d $homedir && echo "true" || echo "false");
        if [ $checkHomeBaseDir == "true" ]; then
            command="$command -d $homedir/$__NAME__";
        else
            globalVar="false";
            echo "useradd: cannot create directory $homedir/$__NAME__";
        fi;
    fi;
    if [ ! -z $EXP_DATE ] ;then
        command="$command -e $EXP_DATE";
    fi;
    if [ ! -z $INACTIVE ] ;then
        command="$command -f $INACTIVE";
    fi;
    if [ ! -z $PGROUP ] ;then
        grp=$PGROUP;
    else
        if [ ! -z $defaultPriGroup ] ;then
            grp=$defaultPriGroup;
        fi;
    fi;
    if [ ! -z $grp ] ;then
        getent group $grp;
        if [ $? -ne 0 ] ;then
            echo "PGROUP=$grp";
            echo "Invalid primary group :- $grp";
            globalVar="false";
        else
            command="$command -g $grp";
        fi;
    fi;
    if [ ! -z $CREATE_HOME_DIR ] && [  $CREATE_HOME_DIR == "true" ] ;then
        command="$command -m";
            if [ ! -z $SKEL_DIR ] ;then
                command="$command -k $SKEL_DIR";
            fi;
    fi;
    if [ ! -z $CREATE_USER_GROUP ] && [  $CREATE_USER_GROUP == "false" ] ;then
        command="$command -n";
    fi;
    if [ ! -z $USER_SHELL ] ;then
        command="$command -s $USER_SHELL";
    else
        if [ ! -z $defaultShell ] ;then
            command="$command -s $defaultShell";
        fi;
    fi;
    if [ ! -z $USID ] && [  $USID -gt 0 ] ;then
        command="$command -u $USID";
        if [ ! -z $UNIQUE_USID ] && [  $UNIQUE_USID == "false" ] ;then
            command="$command -o";
        fi;
    fi;
    if [ $globalVar == "true" ] ;then
        echo "useradd $command $__NAME__";
        useradd $command $__NAME__;
        if [ $? -eq 0 ]; then echo "SUCCESS";
            if [ ! -z "$COMMENTS" ] ; then echo "usermod -c \""$COMMENTS"\" 
$__NAME__";
                usermod -c "$COMMENTS" $__NAME__;
            fi;
        fi;
    fi;
fi;
unset bar COMMENTS HOME_DIR PGROUP grp EXP_DATE SKEL_DIR UNIQUE_USID __NAME__ 
__UID__ CREATE_HOME_DIR;
unset USER_SHELL USID CREATE_USER_GROUP INACTIVE SECONDARYGROUP command 
globalVar name;
unset defaultHomeBaseDir homedir checkHomeBaseDir grp defaultPriGroup;
```

C.2 Updated Sample Script

This is the CreateNativeUser.txt script that has been updated to include the newly added __GID__ attribute. The updated lines are represented in bold font.

```
if [ ! -z "$__UID__" ] ;then
    __NAME__=$__UID__;
else
    __NAME__=$__NAME__;
fi;
if id $__NAME__ > /dev/null 2>&1 ;then
    echo "User already exists";
else
    globalVar="true";
    if [ ! -z $SECONDARYGROUP ] ;then
        command="$command -G $SECONDARYGROUP";
    fi;

    homedir="";
    if [ ! -z $HOME_DIR ] ;then
        homedir=$HOME_DIR;
    else
        if [ ! -z $defaultHomeBaseDir ] ;then
            homedir=$defaultHomeBaseDir;
        fi;
    fi;
    if [ ! -z $homedir ] ;then
        checkHomeBaseDir=$(test -d $homedir && echo "true" || echo "false");
        if [ $checkHomeBaseDir == "true" ]; then
            command="$command -d $homedir/$__NAME__";
        else
            globalVar="false";
            echo "useradd: cannot create directory $homedir/$__NAME__";
        fi;
    fi;
    if [ ! -z $EXP_DATE ] ;then
        command="$command -e $EXP_DATE";
    fi;
    if [ ! -z $INACTIVE ] ;then
        command="$command -f $INACTIVE";
    fi;
    if [ ! -z $PGROUP ] ;then
        grp=$PGROUP;
    else
        if [ ! -z $defaultPriGroup ] ;then
            grp=$defaultPriGroup;
        fi;
    fi;
    if [ ! -z $grp ] ;then
        getent group $grp;
        if [ $? -ne 0 ] ;then
            echo "PGROUP=$grp";
            echo "Invalid primary group :- $grp";
            globalVar="false";
        else
            command="$command -g $grp";
        fi;
    fi;
    if [ ! -z $CREATE_HOME_DIR ] && [  $CREATE_HOME_DIR == "true" ] ;then
```

```
        command="$command -m";
            if [ ! -z $SKEL_DIR ] ;then
                command="$command -k $SKEL_DIR";
            fi;
    fi;
    if [ ! -z $CREATE_USER_GROUP ] && [  $CREATE_USER_GROUP == "false" ] ;then
        command="$command -n";
    fi;
    if [ ! -z $USER_SHELL ] ;then
        command="$command -s $USER_SHELL";
    else
        if [ ! -z $defaultShell ] ;then
            command="$command -s $defaultShell";
        fi;
    fi;
    if [ ! -z $USID ] && [  $USID -gt 0 ] ;then
        command="$command -u $USID";
        if [ ! -z $UNIQUE_USID ] && [  $UNIQUE_USID == "false" ] ;then
            command="$command -o";
        fi;
    fi;
    if [ ! -z $__GID__ ] ;then
        command="$command -g $__GID__";
    fi;
    if [ $globalVar == "true" ] ;then
        echo "useradd $command $__NAME__";
        useradd $command $__NAME__;
        if [ $? -eq 0 ]; then echo "SUCCESS";
            if [ ! -z "$COMMENTS" ] ; then echo "usermod -c \""$COMMENTS"\"
$__NAME__";
                usermod -c "$COMMENTS" $__NAME__;
            fi;
        fi;
    fi;
fi;
unset bar COMMENTS HOME_DIR PGROUP grp EXP_DATE SKEL_DIR UNIQUE_USID __NAME__
__UID__ CREATE_HOME_DIR;
unset USER_SHELL USID CREATE_USER_GROUP INACTIVE SECONDARYGROUP command
globalVar name;
unset defaultHomeBaseDir homedir checkHomeBaseDir grp defaultPriGroup __GID__;
```

D
Files and Directories in the UNIX Connector Package

These are the files and directories on the connector installation package that comprise the UNIX connector.

Table D-1 Files and Directories in the Installation Package

File in the Installation Package Directory	Description
bundle/ org.identityconnectors.genericunix-12.3.0.jar	This JAR file contains the connector bundle.
configuration/GenericUNIX-CI.xml	This XML file contains configuration information that is used during the connector installation process.
Files in the resources directory	Each of these resource bundles contains language-specific information that is used by the connector. During connector installation, these resource bundles are copied to Oracle Identity database. **Note:** A **resource bundle** is a file containing localized versions of the text strings that are displayed on the Administrative and User Console. These text strings include GUI element labels and messages.
Files in the test-utility directory: • example-config.groovy • README • test-utility.jar	These files are used by the testing utility to identify the cause of problems associated with connecting to the target system and performing basic operations on the target system. • The example-config.groovy file is a sample configuration that can be used to set the connection properties of the target system and the connector. • The README file contains instructions to configure and run the testing utility. • The test-utility.jar file contains the class files used by the testing utility.
upgrade/PostUpgradeScriptUnix.sql	This file is used after upgrading the connector. See Upgrading the UNIX Connector for more information.
util/privateKeyGen.sh	This file is used during SSH key-based authentication.
util/sudoers	This file contains the SUDO user specifications and configurations.
xml/genericunix-target-template.xml	This file contains definitions for the connector objects required for creating a Target application. It includes certain details required to connect Oracle Identity Governance with the target system. It also includes configuration details specific to your target system, attribute mappings, correlation rules, and reconciliation jobs.
xml/genericunix-auth-template.xml	This file contains definitions for the connector objects required for creating an Authoritative application. It includes certain details required to connect Oracle Identity Governance with the target system. It also includes configuration details specific to your target system, attribute mappings, correlation rules, and reconciliation jobs.

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

File in the Installation Package Directory	Description
xml/genericunix-pre-config.xml	This XML file contains definitions for the connector objects associated with any non-User objects such as Secondary Groups.
xml/UNIX-ConnectorConfig.xml	This XML file contains definitions for the connector components. These components include the following: • IT resource type • Process form • Process task and adapters (along with their mappings) • Resource object • Provisioning process • Prepopulate rules • Lookup definitions • Scheduled tasks

Index