Oracle® Identity Manager
Connector Guide for Box

Release 11.1.1
E73804-05
May 2020

Oracle Identity Manager Connector Guide for Box, Release 11.1.1

E73804-05

Primary Author: Gowri GR

Contents

2 Deploying the Box Connector

3 Using the Box Connector

4 Extending the Functionality of the Box Connector

A Files and Directories on the Box Connector Installation Media

List of Figures

List of Tables

Preface

This guide describes the connector that is used to integrate Oracle Identity Manager with the Box target system.

Audience

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

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

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

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

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

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

Conventions

The following text conventions are used in this document:

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

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

What's New in Oracle Identity Manager Connector for Box?

This chapter provides an overview of the updates made to the software and documentation for the Box connector in release 11.1.1.5.0.

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

- Software Updates

  This section provides updates made to the connector software. This section also points out the sections of this guide that have been changed in response to each software update.

- Documentation-Specific Updates

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

Software Updates

The following section provides information on software updates:

Software Updates in Release 11.1.1.5.0

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

Documentation-Specific Updates

The following section provides information on documentation-specific updates:

Documentation-Specific Updates in Release 11.1.1.5.0

The following documentation-specific update has been made in revision "5" of this guide:

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

The following documentation-specific update has been made in revision "4" of this guide:

The "Oracle Identity Manager" row of Table 1-1 has been renamed as "Oracle Identity Governance or Oracle Identity Manager" and also updated for Oracle Identity Governance 12*c* (12.2.1.3.0) certification.

The following documentation-specific update has been made in revision "3" of this guide:

The Enabling Logging section has been modified to update the logger name from "ORG.IDENTITYCONNECTORS.GENERICREST" to "ORG.IDENTITYCONNECTORS.BOX".

The following documentation-specific updates have been made in revision "2" of this guide:

- Chapter 5, "Known Issues and Workaround for the Box Connector" has been removed as there are no known issues associated with this connector.
- Title of the section containing SSL configuration information has been corrected.
- Appendix A, "Files and Directories on the Box Connector Installation Media" has been added.
- Editorial corrections have been made throughout the guide.

The following is a documentation-specific update in revision "01" of this guide:

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

1

About the Box Connector

The Box connector integrates Oracle Identity Manager (OIM) with the Box target system.

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

1.1 Introduction to the Box Connector

The Box connector enables you to use Box as a managed (target) source of identity data for Oracle Identity Manager.

Oracle Identity Manager is a centralized identity management solution that provides self service, compliance, provisioning and password management services for applications residing on-premise or on the Cloud. Oracle Identity Manager connectors are used to integrate Oracle Identity Manager with external, identity-aware applications. This guide provides information about the connector that enables you to use Box as a managed (target) resource of Oracle Identity Manager.

> **Note:**
>
> At some places in this guide, the term **target system** has been used to refer to the Box target system.

In the account management (target resource) mode of the connector, data about users created or modified directly on the target system can be reconciled into Oracle Identity Manager. This data is used to provision (allocate) new resources or update resources already assigned to OIM Users. You can use Oracle Identity Manager to provision or update Box resources (that is, accounts) assigned to OIM Users. These provisioning operations performed on Oracle Identity Manager translate into the creation or updates to target system accounts. In addition, the connector aunthenticates to the Box target system using the access token that is provided as a manual input from the user.

1.2 Certified Components for the Box Connector

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

Table 1-1 lists the certified components for this connector.

Table 1-1 Certified Components

Component	Requirement
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) • Oracle Identity Manager 11*g* Release 2 PS2 BP09 • Oracle Identity Manager 11*g* Release 2 PS3 BP06
Target System	Box
Connector Server	11.1.2.1.0
Connector Server JDK	JDK 1.6 or later

1.3 Certified Languages for the Box Connector

These are the languages that the connector supports.

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

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

1.4 Connector Architecture of the Box Connector

The Box connector is implemented by using the Identity Connector Framework (ICF).

The connector enables you to manage accounts on the target system. Managing accounts consists of the following processes:

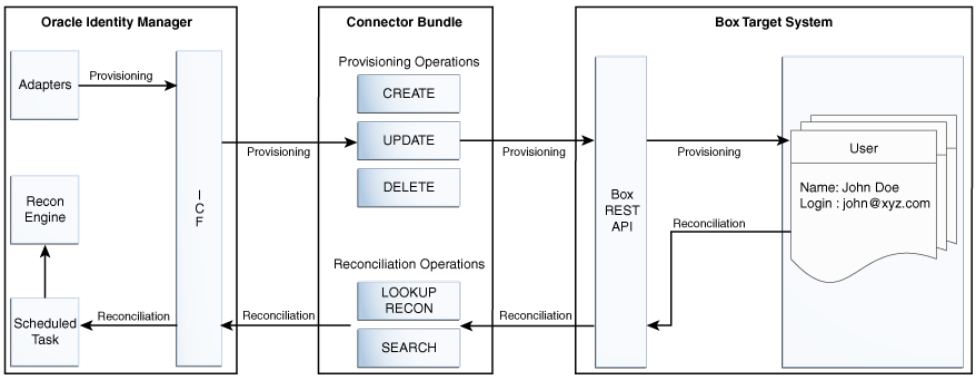
- **Provisioning**

  Provisioning involves creating or updating users on the target system through Oracle Identity Manager. When you allocate (or provision) a Box resource to an OIM User, the operation results in the creation of an account on Box for that user. In the Oracle Identity Manager context, the term provisioning also covers updates made to the target system account through Oracle Identity Manager.

- **Target Resource Reconciliation**

  In target resource reconciliation, data related to newly created and modified target system accounts can be reconciled and linked with existing OIM Users and provisioned resources. A scheduled job is used for reconciliation.

Figure 1-1 Connector Architecture



As shown in this figure, Box is configured as a target resource of Oracle Identity Manager. Through provisioning operations performed on Oracle Identity Manager, accounts are created and updated on the target system for OIM Users. Through reconciliation, account data that is created and updated directly on the target system is fetched into Oracle Identity Manager and stored against the corresponding OIM Users. Identity Connector Framework (ICF) is a component that is required in order to use Identity Connectors. ICF is distributed together with Oracle Identity Manager.

You do not need to configure or modify ICF. During provisioning, the Adapters invoke ICF operation, ICF inturn invokes create operation on Box Identity Connector Bundle and then the bundle calls Box REST API for Provisioning operations. The Box REST API on the target system accepts provisioning data from the bundle, carries out the required operation on the target system, and returns the response from the target system back to the bundle, which passes it to the adapters. During reconciliation, a scheduled task invokes ICF operation, ICF inturn invokes search operation on Box Identity Connector Bundle and then the bundle calls Box REST API for Reconciliation operation. The API extracts user records that match the reconciliation criteria and hands them over through the bundle and ICF back to the scheduled task, which brings the records to Oracle Identity Manager.

Each record fetched from the target system is compared with Box resources that are already provisioned to OIM Users. If a match is found, then the update made to the Box record from the target system is copied to the Box resource in Oracle Identity Manager. If no match is found, then the user ID of the record is compared with the user ID of each OIM User. If a match is found, then data in the target system record is used to provision an Box resource to the OIM User.

The Box Identity Connector Bundle communicates with the Box REST API using the HTTPS protocol.

1.5 Use Cases Supported by the Box Connector

Box is a cloud computing business which provides file-sharing, collaborating, and other tools for working with files that are uploaded to its servers. Box provides dynamic, flexible content management solution which empowers users to share and access content from anywhere, while providing IT enterprise-grade security and oversight into how content moves within their organizations.

The following are some of the most common scenarios in which this connector can be used:

- Automated Admin Token Management

  Today, security is one of the biggest concerns that organizations face while accessing cloud applications. Each cloud application has its own mechanism to ensure that a security breach does not take place. Box uses automated tokens to achieve this. The administrators of the Box Connector are assigned a security token which is required to authenticate and authorize the administrators in order to perform various operations on the application. This token also gets refreshed periodically to minimize risk.

  Oracle Identity Manager Connector for Box provides automated admin token management to ensure that your administrator's security tokens are up-to-date. This functionality ensures that only authenticated and authorized administrators can perform the operations without any delay due to stale or expired tokens.

- Box User Management

  Organizations across the globe are using Box for content sharing. They want their employees to be able to access and share the most up-to-date information across different geographical locations. To achieve this, a Box administrator has to create and grant login to the concerned employees. The Box administrator must also be sure of the complete life cycle of this particular user. It must be ensured that when an employee leaves the organization, they should no longer be able to access sensitive information or content or files using their Box account. Similarly, during an employee's tenure, it has to be ensured that they have access towards files and content which they alone are entitled to use while access must be restricted towards classified files and content.

  Doing this manually for every employee is very cumbersome thus resulting in errors sometimes. Oracle Identity Manager Connector for Box provides user management functionality which enables automation of provisioning and deprovisioning of the users (employees). Whenever a new employee joins the organization, a Box account will automatically be provisioned to them along with appropriate access rights. Likewise, when they leave the organization, the same account will automatically be deactivated. This not only saves time but provides robust security as manual intervention is minimal here.

- Exempt Box User from 2-Step Login Verification

  As the need for enhanced security increases, Box provides an extra layer of security through the use of 2-step verification. This process requires the user to present the following two authenticating pieces of evidence when they log in:

  – Something they know (their Box password)

  – Something they have (an OTP code that is sent to their mobile device)

  The OTP code is sent to the users mobile device as a text message (SMS). In case the user loses their mobile device or cannot access the confirmation codes sent to the mobile device for reasons unknown, the Box Connector provides an option to exempt the user from the 2-Step Login Verification requirement. An exempted user would be able to log in successfully with only the Box password. If you would like to exempt a group of users or the administrator, you can enable the option **Exempt this user from 2-Step login verification** for that particular user.

- User Email Alias Management

  In an organization, a user may have multiple email addresses. For example, in case of acquisitions or mergers, there is a need to manage multiple email addresses for different domains. In such a situation, you may want to add a new email alias for a user who changed their name but left their primary email address the same.

  Email alias allows users to link multiple email addresses to a single Box account for easy management of their important content. Now, any user type can add multiple email addresses to their account and designate one as the primary address, where collaboration invites and Box notifications will be sent. With Oracle Identity Manager Connector for Box, you can manage email aliases and mark any one of them as primary for the user account.

- Box User's Group Membership Management

  Organizations are usually broken down into departments, project teams or other sub-units and with this systematic break-down, there comes a need to grant different teams different levels of access for different content. The group functionality of BOX helps the organization to achieve this, thereby making it easy

to replicate the work or resource breakdown easy in BOX. It also helps in creating new teams along new lines. Groups make this division of labor easy to replicate in Box, and also give the opportunity to create new teams along new lines.

Oracle Identity Manager Connector for Box enables an organization to manage user's group memberships. A user can be a member of one or more groups. Oracle Identity Manager Connector for Box has the capability to enable the IT group to retain visibility into how content is managed and accessed. With monitoring and granular access control capabilities, it can be ensured that only authorized users have access.

One other benefit of collaborating using the Box User's Group Membership Management is that, as new users are added to any specific group, they are automatically eligible to gain access to the content already shared. This means that, such new users can log in and gain access to relevant content required by them to perform their job effectively.

- User and Group Reconciliation

  If a user with an existing Box application (which has other users and groups configured) wants to manage users and group membership, they must initially migrate the pre-existing Box groups into Oracle Identity Manager. The Box Connector facilitates User Reconciliation and Group Lookup Reconciliation to bulk load these users and their group memberships to Oracle Identity Manager respectively.

1.6 Features of the Box Connector

The features of the connector include support for connector server, full reconciliation, and limited reconciliation.

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

1.6.1 Full Reconciliation

In full reconciliation, all records are fetched from the target system to Oracle Identity Governance.

> **Note:**
>
> The connector cannot support incremental reconciliation because of the target system limitation. The target system does not provide a way to filter user records based on the attribute which stores the time at which the account data is created or modified.

In full reconciliation, all records are fetched from the target system to Oracle Identity Governance. See Full Reconciliation for Box Connector.

1.6.2 Limited Reconciliation

You can reconcile records from the target system based on a specified filter criterion.

To limit or filter the records that are fetched into Oracle Identity Governance during a reconciliation run, you can specify the subset of added or modified target system records that must be reconciled. See Limited Reconciliation for Box Connector.

1.6.3 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 want 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 if the bundle works faster when deployed on the same host as the native managed resource.

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

1.6.4 Transformation and Validation of Account Data

You can configure validation of account data that is brought into or sent from Oracle Identity Manager during reconciliation and provisioning.

In addition, you can configure transformation of account data that is brought into Oracle Identity Manager during reconciliation. The following sections provide more information:

- Configuring Validation of Data During Reconciliation and Provisioning
- Configuring Transformation of Data During User Reconciliation

1.7 Lookup Definitions Used During Connector Operations

Lookup definitions used during reconciliation and provisioning are either preconfigured or can be synchronized with the target system.

- Lookup Definition Synchronized with the Target System
- Preconfigured Lookup Definitions for the Box Connector

1.7.1 Lookup Definition Synchronized with the Target System

Lookup field synchronization involves copying additions or changes made to specific fields in the target system to lookup definitions in Oracle Identity Manager.

During a provisioning operation, you use a lookup field on the process form to specify a single value from a set of values. For example, you use the Group Name lookup field to select a group (from a list of group names) to which the user being provisioned must belong to. When you deploy the connector, lookup definitions corresponding to the lookup fields on the target system are created in Oracle Identity Manager. Lookup field synchronization involves copying additions or changes made to the target system lookup fields into the lookup definitions in Oracle Identity Manager.

After you deploy the connector, the Lookup.Box.Groups lookup definition is used as an input source for lookup fields which are automatically created in Oracle Identity Manager.

Lookup fields are populated with values corresponding to the target system that you select for the operation. The Group Name lookup field is populated with values from the Lookup.Box.Groups lookup definition, which is automatically created on Oracle Identity Manager when you deploy the connector. When you perform lookup field synchronization, Groups on the target system are fetched to Oracle Identity Manager and populated in the Lookup.Box.Groups lookup definition. You use the Box Lookup Recon scheduled task to perform lookup field synchronization.

The Lookup.Box.Groups lookup definition holds values of all Groups available on the target system after successful reconciliation. It brings a list of all available and active Groups from Box which are then created in OIM identity console under the Organization tab.

This lookup definition is empty by default and is populated with values fetched from the target system when you run the scheduled jobs for lookup field synchronization.

After lookup field synchronization, data in each of the lookup definitions for lookup field synchronization is stored in the following format:

Code Key:

<IT_RESOURCE_KEY>~<LOOKUP_FIELD_VALUE>

In this format:

- *IT_RESOURCE_KEY* is the numeric code assigned to each IT resource in Oracle Identity Manager.
- *LOOKUP_FIELD_VALUE* is the connector attribute value defined for code.

Sample value: `188~3502898`

In this sample value, `188` is the numeric code assigned to the IT resource associated with the target system and `3502898` is the lookup field value in the target system.

Decode:

<IT_RESOURCE_NAME>~<LOOKUP_FIELD_VALUE>

In this format:

- *IT_RESOURCE_NAME* is the name of the IT resource in Oracle Identity Manager.
- *LOOKUP_FIELD_VALUE* is the connector attribute value defined for code.

Sample value: `BOX~BOX Sales`

In this sample value, `BOX` is the value assigned to the IT resource associated with the target system and `BOX Sales` is the lookup field value in the target system.

1.7.2 Preconfigured Lookup Definitions for the Box Connector

Preconfigured lookup definitions are the other lookup definitions that are created in Oracle Identity Manager when you deploy the connector. These lookup definitions are either prepopulated with values or values must be manually entered in them after the connector is deployed.

The other lookup definitions are as follows:

- Lookup.Box.Configuration
- Lookup.Box.UM.Configuration
- Lookup.Box.UM.ProvAttrMap
- Lookup.Box.UM.ReconAttrMap
- Lookup.Box.Timezone
- Lookup.Box.BooleanValues
- Lookup.Box.Language
- Lookup.Box.Role

1.7.2.1 Lookup.Box.Configuration

The Lookup.Box.Configuration lookup definition holds connector configuration entries that are used during target resource reconciliation and provisioning operations.

> **Note:**
>
> The values for these lookup definitions are preconfigured and cannot be modified.

Table 1-2 lists the default entries in this lookup definition.

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

Code Key	Decode	Description
Bundle Name	org.identityconnectors.genericrest	This entry holds the name of the connector bundle.
Bundle Version	1.0.1115	This entry holds the version of the connector bundle.

Table 1-2 (Cont.) Entries in the Lookup.Box.Configuration Lookup Definition

Code Key	Decode	Description
Connector Name	org.identityconnectors.genericrest.GenericRESTConnector	This entry holds the name of the connector class.
customPayload	"__ACCOUNT__.__GROUP__.UPDATEOP={ \"user\": { \"id\": \"$(__UID__)$\"}, \"group\": { \"id\": \"(id)\" } }","__ACCOUNT__.__GROUP__.CREATEOP={ \"user\": { \"id\": \"$(__UID__)$\"}, \"group\": { \"id\": \"(id)\" } }"	This entry holds the payloads for all operations that are not in the standard format.
httpHeaderAccept	application/json	This holds the accept-type expected from the target system in the header.
httpHeaderContentType	application/json	This holds the content-type expected by the target system in the header.
jsonResourcesTag	"__ACCOUNT__=entries","__GROUP__=entries","__ACCOUNT__.__GROUP__=entries","__ACCOUNT__.email=entries", "__ACCOUNT__.__MEMBERSHIP__.__GROUP__=entries","__ACCOUNT__.__MEMBERSHIP__.email=entries" nameAttributes "__ACCOUNT__.login","__GROUP__.name"	This json tag value is used during reconciliation for parsing multiple entries in a single response payload (ex: Resources).
nameAttributes	"__ACCOUNT__.login","__GROUP__.name"	This entry holds the name attribute for all the objects that are handled by this connector. For example, for the __ACCOUNT__ object class that it used for User accounts, the name attribute is login.
opTypes	"__ACCOUNT__.CREATEOP=POST","__ACCOUNT__.UPDATEOP=PUT","__ACCOUNT__.DELETEOP=DELETE", "__ACCOUNT__.__GROUP__.UPDATEOP=POST","__ACCOUNT__.__GROUP__.DELETEOP=DELETE"," __ACCOUNT__.email.UPDATEOP=POST","__ACCOUNT__.email.DELETEOP=DELETE"	This entry specifies the HTTP operation type for each object class supported by the connector. Values are comma separated and are in the following format: Format: *OBJ_CLASS.OP=HTTP_OP* In this format, OBJ_CLASS is the connector object class, OP is the connector operation (for example, CreateOp, UpdateOp, SearchOp), and HTTP_OP is the HTTP operation (GET, PUT, or POST).

Table 1-2 (Cont.) Entries in the Lookup.Box.Configuration Lookup Definition

Code Key	Decode	Description
relURIs	"__ACCOUNT__.CREATEOP=/users","__ACCOUNT__.UPDATEOP=/users/$(__UID__)$","__ACCOUNT__.SEARCHOP=/users/$(FilterSuffix)$","__ACCOUNT__.DELETEOP=/users/$(__UID__)$","__GROUP__.SEARCHOP=/groups/$(FilterSuffix)$","__ACCOUNT__.__GROUP__.CREATEOP=/group_memberships","__ACCOUNT__.__GROUP__.UPDATEOP=/group_memberships","__ACCOUNT__.__GROUP__.SEARCHOP=/users/$(__UID__)$/memberships","__ACCOUNT__.__GROUP__.DELETEOP=/group_memberships/$(__MEMBERSHIP__.id)$","__ACCOUNT__.__MEMBERSHIP__.__GROUP__.SEARCHOP=/users/$(__UID__)$/memberships","__ACCOUNT__.email.UPDATEOP=/users/$(__UID__)$/email_aliases","__ACCOUNT__.email.SEARCHOP=/users/$(__UID__)$/email_aliases","__ACCOUNT__.email.DELETEOP=/users/$(__UID__)$/email_aliases/$(__MEMBERSHIP__.id)$","__ACCOUNT__.__MEMBERSHIP__.email.SEARCHOP=/users/$(__UID__)$/email_aliases","__ACCOUNT__.role.SEARCHOP=/users/$(__UID__)$?fields=role","__ACCOUNT__.is_sync_enabled.SEARCHOP=/users/$(__UID__)$?fields=is_sync_enabled","__ACCOUNT__.is_exempt_from_login_verification.SEARCHOP=/users/$(__UID__)$?fields=is_exempt_from_login_verification"	This entry holds the relative URL of every object class supported by this connector and the connector operations that can be performed on these object classes. For example, the __ACCOUNT__.UPDATEOP=/users/$(__UID__)$ is the relative URL for all update provisioning operations performed on the __ACCOUNT__ object class.

Table 1-2 (Cont.) Entries in the Lookup.Box.Configuration Lookup Definition

Code Key	Decode	Description
specialAttributeHandling	"__ACCOUNT__.__GROUP__.CREATEOP=SINGLE","__ACCOUNT__.__GROUP__.UPDATEOP=SINGLE","__ACCOUNT__.email.CREATEOP=SINGLE",	This entry specifies the special attributes whose values should be sent to the target system one by one ("SINGLE"). Values should be comma separated.
	"__ACCOUNT__.email.UPDATEOP=SINGLE","__ACCOUNT__.role.SEARCHOP=SINGLE","__ACCOUNT__.is_exempt_from_login_verification.SEARCHOP=SINGLE",	Format: objectClass.attributeName.operartion=SINGLE
	"__ACCOUNT__.is_sync_enabled.SEARCHOP=SINGLE"	
specialAttributeTargetFormat	"__ACCOUNT__.__GROUP__=group","__ACCOUNT__.__MEMBERSHIP__.__GROUP__=group.id"	This entry lists the format in which a special attribute is present in the target system endpoint. For example, the alias attribute will be present as aliases.alias in the target system endpoint. Values are comma separated and are presented in the following format:
		OBJ_CLASS.ATTR_NAME=TARGET_FORMAT
statusAttributes	__ACCOUNT__.status	This entry lists the name of the target system attribute that holds the status of an account. For example, for the __ACCOUNT__ object class that it used for User accounts, the status attribute is accountEnabled.
statusDisableValue	inactive	This entry holds the value of the status attribute in the target system which represents the disable value.
statusEnableValue	active	This entry holds the value of the status attribute in the target system which represents the enable value.
uidAttributes	"__ACCOUNT__.id","__GROUP__.id"	This entry holds the UID attribute for the object which are handled by the connectors. For example for User accounts the uid attribute is id.

Table 1-2 (Cont.) Entries in the Lookup.Box.Configuration Lookup Definition

Code Key	Decode	Description
User Configuration Lookup	Lookup.Box.UM.Configuration	This entry holds the name of the lookup Lookup.Box.UM.Configuration definition that stores configuration information used during user management operations.
enableEmptyString	true	This entry holds the configuration value. If this configuration is set to `true`, the connector will send an empty string instead of null to the target system when any attribute of the Box account of Oracle Identity Manager user is updated with a blank value.

1.7.2.2 Lookup.Box.UM.Configuration

The Lookup.Box.UM.Configuration lookup definition holds configuration entries that are specific to the user object type. This lookup definition is used during user management operations in the target resource mode.

Table 1-3 lists the entries in this lookup definition.

Table 1-3 Entries in the Lookup.Box.UM.Configuration Lookup Definition

Code Key	Decode	Description
Provisioning Attribute Map	Lookup.Box.UM.ProvAttrMap	This entry holds the name of the lookup definition that stores attribute mappings between Oracle Identity Manager and the target system. This lookup definition is used during user provisioning operations
Recon Attribute Map	Lookup.Box.UM.ReconAttrMap	This entry holds the name of the lookup definition that stores attribute mappings between Oracle Identity Manager and the target system. This lookup definition is used during reconciliation

1.7.2.3 Lookup.Box.UM.ProvAttrMap

The Lookup.Box.UM.ProvAttrMap lookup definitions hold mappings between process form fields and target system attributes.

This lookup definition is preconfigured and used during provisioning. Table 1-11 lists the default entries.

You can add entries in this lookup definitions if you want to map new target system attributes for provisioning. See Adding New User Attributes for Provisioning.

1.7.2.4 Lookup.Box.UM.ReconAttrMap

The Lookup. Box.UM.ReconAttrMap lookup definition holds mappings between resource object fields and target system attributes.

This lookup definition is preconfigured and used during target resource reconciliation. Table 1-8 lists the default entries. You can add entries in this lookup definitions if you want to map new target system attributes for target resource reconciliation. See Adding New User Attributes for Reconciliation.

1.7.2.5 Lookup.Box.Timezone

The Lookup.Box.Timezone lookup definition holds information about timezones that you can assign to a target system account that you create through Oracle Identity Manager. This is a static lookup definition.

You must populate the entries of this lookup definition manually. The following is the format of the Code Key and Decode values in this lookup definition:

- **Code Key:** Timezone value as expected by the target system.
- **Decode:** Value which will be displayed to User on OIM form. This can be same as the decode key value or can modified by the user for better readability purposes.

Table 1-4 lists default entries in this lookup definition.

Table 1-4 Entries in the Lookup.Box.Timezone Lookup Definition

Code Key	Decode
America/Chicago	America/Chicago
America/Los_Angeles	America/Los_Angeles
America/New_York	America/New_York
Asia/Calcutta	Asia/Calcutta
Asia/Hong_Kong	Asia/Hong_Kong
Australia/Melbourne	Australia/Melbourne
Australia/Sydney	Australia/Sydney
Europe/London	Europe/London

1.7.2.6 Lookup.Box.BooleanValues

The Lookup.Box.BooleanValues lookup definition maps boolean values that are used for some of the fields in the target system with the corresponding boolean values to be displayed in the fields of the OIM User form.

Table 1-5 lists default entries in this lookup definition.

Table 1-5 Entries in the Lookup.Box.BooleanValues Lookup Definition

Code Key	Decode
0	False
1	True

1.7.2.7 Lookup.Box.Language

The Lookup.Box.Language lookup definition holds a list of languages that you can assign as a preferred language for a target system account that you create through Oracle Identity Manager. This is a static lookup definition.

You must populate the entries of this lookup definition manually. The following is the format of the Code Key and Decode values in this lookup definition:

- **Code Key:** 2-letter ISO code for a language
- **Decode:** Language name

Table 1-6 lists the default entry in this lookup definition.

Table 1-6 Entry in the Lookup.Box.Language Lookup Definition

Code Key	Decode
en	English

1.7.2.8 Lookup.Box.Role

The Lookup.Box. Role lookup definition holds information about roles that you can select for a target system user account that you create through Oracle Identity Manager. This is a static lookup definition.

These Roles can be seen as a Lookup value in the Box Form. To add new attributes that you want the connector to handle, we can add new entries to this lookup definition using the Design console. See Adding New User Attributes for Reconciliation for more information about adding a new user attribute during reconciliation. When you go to the Lookup definition, you can see the Add and the Delete buttons. These can help you to add and delete entries. See Adding New User Attributes for Provisioning for more information about adding a new user attribute during provisioning.

You must populate the entries of this lookup definition manually. The following is the format of the Code Key and Decode values in this lookup definition:

- **Code Key:** Value as expected by the target system.
- **Decode:** Value which will be displayed to User on OIM form. This can be same as the decode key value or can modified by the user.

Table 1-7 holds default entries for this lookup definition.

Table 1-7 Entries in the Lookup.Box.Role Lookup Definition

Code Key	Decode
coadmin	Co-Admin
user	User

1.8 Connector Objects Used During Target Resource Reconciliation

Target resource reconciliation involves fetching data about newly created or modified accounts on the target system and using this data to add or modify resources assigned to OIM Users.

The Box Resource User Reconciliation scheduled job is used to initiate a reconciliation run. This scheduled job is discussed in Reconciliation Scheduled Jobs for Box Connector.

> **See Also:**
>
> Managing Reconciliation in *Oracle Fusion Middleware Administering Oracle Identity Manager* for generic information about connector reconciliation

This section discusses the following topics :

- User Fields for Target Resource Reconciliation
- Reconciliation Rules for the Box Connector
- Reconciliation Action Rules for the Box Connector

1.8.1 User Fields for Target Resource Reconciliation

The Lookup. Box.UM.ReconAttrMap lookup definition holds mappings between resource object fields and target system attributes. This lookup definition is used for performing target resource user reconciliation runs.

In this lookup definition, entries are in the following format:

- **Code Key:** Reconciliation field of the resource object
- **Decode:** Name of the target system attribute

Table 1-8 lists the default entries in this lookup definition.

Table 1-8 Entries in the Lookup. Box.UM.ReconAttrMap Lookup Definition

Code Key	Decode
Address	address
Aliases~Alias	email

Table 1-8 (Cont.) Entries in the Lookup. Box.UM.ReconAttrMap Lookup Definition

Code Key	Decode
Groups~Group Name[LOOKUP]	__GROUP__~__GROUP__~id
ID	__UID__
Job Title	job_title
Language	language
Login	__NAME__
Name	name
Phone	phone
Space Amount	space_amount
Status	__ENABLE__
Timezone	timezone
Role	role
Enable Sync	is_sync_enabled
Exempt From Login Verification	is_exempt_from_login_verification

1.8.2 Reconciliation Rules for the Box Connector

Reconciliation rules for target resource reconciliation are used by the reconciliation engine to determine the identity to which Oracle Identity Manager must assign a newly discovered account on the target system.

The following sections provide information about the reconciliation rules for this connector:

- Reconciliation Rule for Users
- Viewing Reconciliation Rules in Design Console

1.8.2.1 Reconciliation Rule for Users

Rule name: Box User Recon Rule

Rule element: User Login Equals Tokenize (Login).

In this rule component:

- User Login is the attribute which is present in the OIM User form.
- Tokenize (Login) where Login is the name attribute which is unique field of the Box account.

1.8.2.2 Viewing Reconciliation Rules in Design Console

> **Note:**
>
> Perform the following procedure only after the connector is deployed.

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Development Tools** and then double-click **Reconciliation Rules.**
3. Search for and open **Box User Recon Rule**.

1.8.3 Reconciliation Action Rules for the Box Connector

Reconciliation action rules specify actions that must be taken depending on whether or not matching Box resources or OIM Users are found when the reconciliation rule is applied.

> **Note:**
>
> No action is performed for rule conditions that are not predefined for this connector. You can define your own action rule for such rule conditions.

The following sections provide information about the action rules for this connector:

- Reconciliation Action Rules
- Viewing Reconciliation Action Rules Reconciliation in Design Console

1.8.3.1 Reconciliation Action Rules

Table 1-9 lists the action rules for target resource reconciliation.

Table 1-9 Action Rules for Reconciliation for Target System

Rule Condition	Action
No Matches Found	None
One Entity Match Found	Establish Link
One Process Match Found	Establish Link

1.8.3.2 Viewing Reconciliation Action Rules Reconciliation in Design Console

After you deploy the connector, you can view the reconciliation action rules for user reconciliation by performing the following steps:

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Resource Management** and double-click **Resource Objects.**
3. Open the **Box User** resource object.
4. Click the **Object Reconciliation** tab, and then click the **Reconciliation Action Rules** tab. The Reconciliation Action Rules tab displays the action rules defined for this connector.

1.9 Connector Objects Used During Provisioning

Provisioning involves creating or modifying user data on the target system through Oracle Identity Manager.

This section contains the following topics:

- Provisioning Functions
- User Fields for Provisioning

1.9.1 Provisioning Functions

These are the supported provisioning functions and the adapters that perform these functions for the Box connector.

Table 1-10 lists the supported provisioning functions and the adapters that perform these functions. The Adapter column gives the name of the adapter that is used when the function is performed.

> **See Also:**
>
> Types of Adapters in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for generic information about process tasks and adapters

Table 1-10 User Provisioning Functions

Function	Adapter
Create User	adpBOXCREATEOBJECT
Enable User	adpBOXENABLE
Disable User	adpBOXDISABLE
Update User	adpBOXUPDATEOBJECTTASK
Delete User	adpBOXDELETEOBJECT
Add Child table values	adpBOXADDCHILDOBJECT
Remove Child table values for user	adpBOXREMOVECHILDOBJECT
Update Child table values	adpBOXUPDATECHILDDATA

1.9.2 User Fields for Provisioning

The Lookup.Box.UM.ProvAttrMap lookup definition maps process form fields with Box fields. This lookup definition is used for performing user provisioning operations.

In this lookup definition, entries are in the following format:

Code Key: Name of the process form field.

Decode: Name of the target system attribute.

Table 1-11 lists the default entries in this lookup definition.

Table 1-11 Entries in the Lookup.Box.UM.ProvAttrMap Lookup Definition

Code Key	Decode
Address	address
Enable Sync	is_sync_enabled
Exempt From Login Verification	is_exempt_from_login_verification
ID	__UID__
Job Title	job_title
Language	language
Login	__NAME__
Name	name
Phone	phone
Role	role
Space Amount	space_amount
Timezone	timezone
UD_BOX_UAL~Alias	email
UD_BOX_UGP~Group Name[LOOKUP]	__GROUP__~__GROUP__~id

1.10 Roadmap for Deploying and Using the Connector

The following is the organization of information in the rest of this guide:

- Deploying the Box Connector describes procedures that you must perform on Oracle Identity Manager and the target system during each stage of connector deployment.
- Using the Box Connector describes guidelines on using the connector and the procedure to configure reconciliation runs and perform provisioning operations.
- Extending the Functionality of the Box Connector describes procedures that you can perform if you want to extend the functionality of the connector.
- Files and Directories on the Box Connector Installation Media lists the files and directories that comprise the connector installation media.

2
Deploying the Box Connector

The procedure to deploy the connector is divided across three stages namely preinstallation, installation, and postinstallation.

The following topics provide details on these stages:

- Preinstallation
- Installation
- Postinstallation

2.1 Preinstallation

Preinstallation for the Box connector involves performing a series of tasks on the target system.

Preinstallation involves the following tasks:

1. Create a Box service account on the target system to manage users on Box through Oracle Identity Manager.

   Ensure that the account is created with the **Co-Admin** role and **Manage Users** and **Manage Groups** administrative privileges.
2. Register the client application of the connector to provide a secure sign-in and authorization for your services.
3. On successful registration of the client application, configure your newly registered application to obtain the Client ID and Client Secret values.

   These values are required to generate access and refresh tokens for your application.
4. Generate access and refresh tokens using the Box service account created in Step 1 and the Client ID and Client Secret values obtained in Step 3.

   Access and refresh tokens must be generated manually for the first time. These tokens are used while configuring the IT Resource in IT Resource Parameters. Access and refresh tokens expire in 60 minutes and 60 days respectively. To avoid this, the Box Update Access Token scheduled job runs a scheduler to renew these values in a periodic manner. This scheduled job is discussed later.

   > **Note:**
   >
   > In case the Box Update Access Token scheduled job fails to run as expected and both access and refresh tokens expire, you must perform this procedure to generate a new pair of access and refresh tokens. Once these values are obtained, you must configure the IT Resource in IT Resource Parameters

The detailed instructions for performing these preinstallation tasks are available in the Box product documentation. For more information, visit the Box website at https://docs.box.com/docs.

2.2 Installation

You must install the connector in Oracle Identity Manager. If necessary, you can also deploy the connector in a Connector Server.

The following topics provide details on installing the Box connector:

- Understanding Installation of the Box Connector
- Running the Connector Installer
- Configuring the IT Resource for the Target System

2.2.1 Understanding Installation of the Box Connector

You can run the connector code either locally in Oracle Identity Manager or remotely in a Connector Server.

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

- Run the connector code locally in Oracle Identity Manager. In this scenario, you deploy the connector in Oracle Identity Manager. Deploying the connector in Oracle Identity Manager involves performing the procedures described in Running the Connector Installer and Configuring the IT Resource for the Target System.
- Run the connector code remotely in a Connector Server. In this scenario, you deploy the connector in Oracle Identity Manager, and then, deploy the connector bundle in a Connector Server. See Using an Identity Connector Server in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about installing, configuring, and running the Connector Server, and then installing the connector in a Connector Server.

2.2.2 Running the Connector Installer

When you run the Connector Installer, it automatically copies the connector files to directories in Oracle Identity Manager, imports connector XML files, and compiles adapters used for provisioning.

To run the Connector Installer, perform the following procedure:

1. Copy the contents of the connector installation media into the following directory:

   OIM_HOME/server/ConnectorDefaultDirectory

2. Log in to Oracle Identity System Administration.
3. In the left pane, under Provisioning Configuration, click **Manage Connector.**
4. In the Manage Connector page, click **Install.**
5. From the **Connector List** list, select **Box Connector *RELEASE_NUMBER*.** This list displays the names and release numbers of connectors whose installation files you copy into the default connector installation directory: *OIM_HOME*/server/

`ConnectorDefaultDirectory`. If you have copied the installation files into a different directory, then:

a. In the Alternative Directory field, enter the full path and name of that directory.

b. To repopulate the list of connectors in the Connector List list, click **Refresh.**

c. From the Connector List list, select **Box Connector *RELEASE_NUMBER*.**

6. Click **Load**.

7. To start the installation process, click **Continue.** In a sequence, the following tasks are automatically performed:

   a. Connector library configuration.

   b. Import of the connector XML files (by using the Deployment Manager).

   c. Adapter compilation.

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

   a. Retry the installation by clicking **Retry.**

   b. Cancel the installation and begin the procedure from Step 3.

8. If all three tasks of the connector installation process are successful, then a message indicating successful installation is displayed.

9. Click **Exit** to close the installation page.

When you run the Connector Installer, it copies the connector files and external code files to destination directories on the Oracle Identity Manager host computer. These files are listed in Files and Directories on the Box Connector Installation Media.

2.2.3 Configuring the IT Resource for the Target System

An IT resource for your target system is created after you install the connector. You configure this IT resource to let the connector connect Oracle Identity Manager with your target system.

This section provides information about the following topics:

- IT Resource Parameters
- Specifying Values for the IT Resource Parameters

2.2.3.1 IT Resource Parameters

An IT resource is composed of parameters that store connection and other generic information about a target system. Oracle Identity Manager uses this information to connect to a specific installation or instance of your target system.

The list of IT resource parameters for this connector can be grouped into the following categories:

- Connection Parameters
- Authentication Parameters

Table 2-1 lists connection parameters.

Table 2-1 Connection IT Resource Parameters

Parameter	Description
Host	Host name or IP address of the computer hosting the target system. Sample value: `api.box.com/2.0`
Port	Port number at which the target system is listening. Sample value: `443`
authenticationType	Type of authentication used by your target system. The Box target system uses manual input of access token and refresh token for OAuth2.0 authentication. Default value: `other` Do *not* modify the value of the parameter.
clientId	Enter the client identifier (a unique string) issued by the authorization server to your client application during the registration process. You would have obtained the client ID while configuring the newly registered application as a part of preinstallation. Sample value: `XDETh0r2eWuULCDVt`
clientSecret	Enter the secret key used to authenticate the identity of your client application. You would have obtained the secret key while configuring the newly registered application as a part of preinstallation. Sample value: `c1ZsdZisTOoYN5NITirarIDepDkiJTGHdzNFT0m`
Configuration Lookup	Name of the lookup definition that stores configuration information used during reconciliation and provisioning operations. Default value: `Lookup.Box.Configuration`
Connector Server Name	If you have deployed the Box connector in a Java Connector Server, then enter the name of the IT resource for the Connector Server.
sslEnabled	If the target system requires SSL connectivity, then set the value of this parameter to true. Otherwise set the value to false. Default value:`true`
proxyHost	Name of the proxy host used to connect to an external target system. Sample value: `proxy.box.com`
proxyPort	Proxy port number. Sample value: `80`
proxyUser	Proxy user name of the target system user account that Oracle Identity Manager uses to connect to the target system.

Table 2-1 (Cont.) Connection IT Resource Parameters

Parameter	Description
proxyPassword	Password of the proxy user ID of the target system user account that Oracle Identity Manager uses to connect to the target system.

Table 2-2 lists authentication parameters.

Table 2-2 Authentication IT Resource Parameters

Parameter	Description
clientId	The client identifier issued to the client during the registration process. Sample value: `XDETh0r2eWuULCDVt`
clientSecret	The client secret identifier issued to the client during the registration process. Sample value: `c1ZsdZisTOoYN5NITirarIDepDkiJTGHdzNFT0m`
customAuthHeaders	Takes access token and refresh token values. Sample value 1: `“access_token=U7CWQIfwxDHhgtcz5NSXUs9udzvcxJOBy”` Sample value 2: `“refresh_token=j2SJIEXMDmTTYmHHMbmproppQAy58xeepe1CAasdggkgzSnk3vstlBLV9pKV20hUR”`

2.2.3.2 Specifying Values for the IT Resource Parameters

The IT resource for the target system contains connection information about the target system. Oracle Identity Manager uses this information during provisioning and reconciliation.

The Box IT resource is automatically created when you run the Connector Installer. You must specify values for the parameters of the IT resource. To specify values:

1. Log in to Oracle Identity System Administration.
2. In the left pane, under Configuration, click **IT Resource.**
3. In the **IT Resource Name** field on the Manage IT Resource page, enter `Box` and then click **Search.**
4. Click **Edit** for the IT resource.
5. From the list at the top of the page, select **Details and Parameters.**
6. Specify values for the parameters of the Box IT Resource. IT Resource Parameters describes each parameter.
7. To save the values, click **Update.**

2.3 Postinstallation

Postinstallation for the Box connector involves configuring Oracle Identity Manager, enabling logging to track information about all connector events, and configuring SSL. It also involves performing some optional configurations such as localizing the user interface.

The postinstallation steps are divided across the following sections:

- Configuring Oracle Identity Manager
- Localizing Field Labels in UI Forms
- Clearing Content Related to Connector Resource Bundles from the Server Cache
- Managing Logging for the Box Connector
- Configuring SSL for the Box Connector

2.3.1 Configuring Oracle Identity Manager

You must create an UI form and an application instance for the resource against which you want to perform reconciliation and provisioning operations. In addition, you must run entitlement and catalog synchronization jobs.

These procedures are described in the following sections:

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

2.3.1.1 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 and Deactivating a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

2.3.1.2 Creating a New UI Form

You can use Form Designer in Oracle Identity System Administration to create and manage application instance forms. See Creating Forms By Using the Form Designer in *Oracle Fusion Middleware Administering Oracle Identity Manager*.

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

2.3.1.3 Creating an Application Instance

Perform the following steps to create an application instance:

1. In the left pane of the System Administration console, under Configuration, click **Application Instances.** The Application Instances page is displayed.
2. From the Actions menu, select **Create.** Alternatively, click **Create** on the toolbar. The Create Application Instance page is displayed.
3. Specify values for the following fields:
   - **Name:** The name of the application instance.
   - **Display Name:** The display name of the application instance.
   - **Description:** A description of the application instance.
   - **Resource Object:** The resource object name. Click the search icon next to this field to search for and select the **Box** User.
   - **IT Resource Instance:** The IT resource instance name. Click the search icon next to this field to search for and select **Box**.
   - **Form:** Select the form name (created in Creating a New UI Form).
4. Click **Save.** The application instance is created.
5. Publish the application instance to an organization to make the application instance available for requesting and subsequent provisioning to users. See Publishing an Application Instance to Organizations in *Oracle Fusion Middleware Administering Oracle Identity Manager*.

2.3.1.4 Publishing a Sandbox

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

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

2.3.1.5 Harvesting Entitlements and Sync Catalog

To harvest entitlements and sync catalog:

1. Run the scheduled jobs for lookup field synchronization listed in Scheduled Job for Lookup Field Synchronization for Box Connector.
2. Run the Entitlement List scheduled job to populate Entitlement Assignment schema from child process form table.

3. Run the Catalog Synchronization Job scheduled job.

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

2.3.1.6 Updating an Existing Application Instance with a New Form

For any changes you do in the Form Designer, 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 a sandbox and activate it. See Creating a Sandbox and Activating and Deactivating a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.
2. Create a new UI form for the resource. See Creating Forms By Using the Form Designer in *Oracle Fusion Middleware Administering Oracle Identity Manager*.
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 Publishing a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

2.3.2 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. Resource bundles are available in the connector installation media.

To localize a field label that is added to the UI forms:

1. Log in to Oracle Enterprise Manager.
2. In the left pane, expand Application Deployments and then select **oracle.iam.console.identity.sysadmin.ear**.
3. In the right pane, from the Application Deployment list, select **MDS Configuration**.
4. On the MDS Configuration page, click **Export** and save the archive to the local computer.
5. Extract the contents of the archive, and open the following file in a text editor:
   `SAVED_LOCATION\xliffBundles\oracle\iam\ui\runtime\BizEditorBundle_en_US.xlf`
6. Edit the BizEditorBundle_en_US.xlf file in the following manner:
   a. Search for the following text:

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

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

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

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

```
<trans-unit id="$
{adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResource
Bundle']
['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.us
erEO.UD_BOX_LOGIN__c_description']}">
<source>Login</source>
<target/>
</trans-unit>
<trans-unit
id="sessiondef.oracle.iam.ui.runtime.form.model.BoxForm.entity.BoxFo
rmEO.UD_BOX_LOGIN__c_LABEL">
<source>Login</source>
<target/>
</trans-unit>
```

d. Open the resource file from the connector package, for example Box_ja.properties, and get the value of the attribute from the file, for example,

```
global.udf.UD_BOX_LOGIN=\u30ED\u30B0\u30A4\u30F3
```

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

```
<trans-unit id="$
{adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResource
Bundle']
['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.us
erEO.UD_BOX_LOGIN__c_description']}">
<source>Login</source>
<target>\u30ED\u30B0\u30A4\u30F3</target>
</trans-unit>
<trans-unit
id="sessiondef.oracle.iam.ui.runtime.form.model.BoxForm.entity.BoxFo
rmEO.UD_BOX_LOGIN__c_LABEL">
<source>Login</source>
<target>\u30ED\u30B0\u30A4\u30F3</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_ja.xlf.

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

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

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

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

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

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

1. In a command window, switch to the *`OIM_HOME`*`/server/bin directory`.
2. Enter one of the following commands:
   - On Microsoft Windows: PurgeCache.bat All
   - On UNIX: PurgeCache.sh All

> **Note:**
>
> You can use the PurgeCache utility to purge the cache for any content category. Run PurgeCache.bat CATEGORY_NAME on Microsoft Windows or PurgeCache.sh CATEGORY_NAME on UNIX. The CATEGORY_NAME argument represents the name of the content category that must be purged.
>
> For example, the following commands purge Metadata entries from the server cache:
>
> - PurgeCache.bat MetaData
> - PurgeCache.sh MetaData
>
> Before running the PurgeCache utility, ensure the WL_HOME and JAVA_HOME environment variables are set.

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

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

In this format:

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

2.3.4 Managing Logging for the Box Connector

Oracle Identity Manager uses 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

2.3.4.1 Understanding Log Levels

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

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

- SEVERE.intValue()+100

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

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

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

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

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

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

These message types are mapped to ODL message type and level combinations as shown in Table 2-3.

Table 2-3 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	TRACE16
FINEST	TRACE32

The configuration file for OJDL is logging.xml is located at the following path:
DOMAIN_HOME/config/fmwconfig/servers/OIM_SERVER/logging.xml

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

2.3.4.2 Enabling Logging

To enable logging in Oracle WebLogic Server:

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

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

   b. Replace both occurrences of [LOG_LEVEL] with the ODL message type and level combination that you require. Table 2-3 lists the supported message type and level combinations. Similarly, replace [FILE_NAME] with the full path and

name of the log file in which you want log messages to be recorded. The following blocks show sample values for **[LOG_LEVEL]** and **[FILE_NAME]:**

```
<log_handler name='BOX-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\o im_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.BOX" level="NOTIFICATION:
1"useParentHandlers="false">
<handler name="BOX-handler"/>
<handler name="console-handler"/>
</logger>
```

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

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

     ```
     set WLS_REDIRECT_LOG=FILENAME
     ```

   - For UNIX:

     ```
     export WLS_REDIRECT_LOG=FILENAME
     ```

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

2.3.5 Configuring SSL for the Box Connector

Configure SSL to secure data communication between Oracle Identity Manager and the target system.

1. Obtain the SSL certificate by obtaining the public key certificate of the target system.
2. Copy the public key certificate of the target system to the computer hosting Oracle Identity Manager.
3. Run the following keytool command to import the public key certificate into the identity key store in Oracle Identity Manager:

```
keytool -import -keystore WEBLOGIC_HOME/server/lib/DemoTrust.jks -file
CERT_FILE_NAME -storepass PASSWORD
```

In this command:

- *CERT_FILE_NAME* is the full path and name of the certificate file
- *PASSWORD* is the password of the keystore.

The following is a sample value for this command:

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

> **Note:**
>
> Change the parameter values passed to the keytool command according to your requirements. Ensure that there is no line break in the keytool arguments.

3
Using the Box Connector

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

This chapter contains the following topics:

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

3.1 Scheduled Job for Lookup Field Synchronization for Box Connector

Scheduled jobs for lookup field synchronization fetch the most recent values from specific fields in the target system to lookup definitions in Oracle Identity Manager. These lookup definitions are used as an input source for lookup fields in Oracle Identity Manager.

The Box Group Lookup Reconciliation scheduled job is used to reconcile group lookup data from the target system in the target resource (account management) mode of the connector. The values that are fetched by this scheduled job are populated in the Lookup.Box.Groups lookup definition.

Table 3-1 describes attributes of the Box Group Lookup Reconciliation scheduled job. The procedure to configure scheduled jobs is described later in this guide.

Table 3-1 Attributes of the Box Group Lookup Reconciliation Scheduled Job

Attribute	Description
Code Key Attribute	Name of the connector 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:`UID`
Decode Attribute	Name of the connector 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`
IT Resource Name	Name of the IT resource for the target system installation from which you want reconcile user records. Default value: `Box`
Lookup Name	Enter the name of the lookup definition in Oracle Identity Manager that must be populated with values fetched from the target system. Default value: `Lookup.Box.Groups` If you create a copy of the Lookup.Box.Groups definition, then enter the name of that new lookup definition as the value of the Lookup Name attribute.
Object Type	This attribute is used to perform reconciliation of specified object type. As per the Scheduled job select the applicable object type. Default value: `__GROUP__`

3.2 Configuring Reconciliation for Box Connector

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

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

- Full Reconciliation for Box Connector
- Limited Reconciliation for Box Connector
- Reconciliation Scheduled Jobs for Box Connector

3.2.1 Full Reconciliation for Box Connector

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

After you deploy the connector, you must first perform full reconciliation. To perform a full reconciliation run, ensure that no value is specified for the Filter attribute of the scheduled job for reconciling users.

3.2.2 Limited Reconciliation for Box Connector

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

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

The scheduled job provides a Filter Suffix parameter that allows you to use any of the Box resource attributes to filter the target system records. You can perform limited reconciliation by creating filters for the reconciliation module. For detailed information about the various filter syntax that are supported, refer the Box documentation.

For the Filter Suffix attribute on the scheduled job, following are sample values that can be provided:

- ?filter_term=sand

  In the above sample, **sand** is specified after the **?filter_term=** syntax in the filter suffix attribute. This returns all users starting with the term sand in either the name or the login values.

  Similarly, any value specified after the **?filter_term=** syntax returns users whose name or login begins with the string value specified in the filter syntax field.

- /181216415

  In the above sample, **181216415** is specified after the **/** syntax in the filter suffix attribute. This returns all users records whose UID matches 181216415.

  Similarly, any value specified after the **/** syntax returns users whose UID attributes which is equal to the string specified in the filter syntax field.

3.2.3 Reconciliation Scheduled Jobs for Box Connector

When you run the Connector Installer, reconciliation scheduled jobs are automatically created in Oracle Identity Manager. You must configure these scheduled jobs to suit your requirements by specifying values for its attributes.

This section discusses the following scheduled jobs that you can configure for reconciliation:

- Box User Reconciliation
- Box Update Access Token Job

3.2.3.1 Box User Reconciliation

You use the Box User Reconciliation scheduled job to reconcile user account data from the target system.

Table 3-3 describes the attributes of this scheduled job.

Table 3-2 Attributes of the Box User Reconciliation Scheduled Task

Attribute	Description
Filter Suffix	Enter the search filter for fetching user records from the target system during a reconciliation run. See Limited Reconciliation for Box Connector.
Latest Token	This attribute holds the value of the attribute that is specified as the value of the Incremental Recon Attribute. The Latest Token attribute is used for internal purposes. By default, this value is empty. Do *not* enter a value for this attribute. The reconciliation engine automatically enters a value in this attribute. Sample value: `1354753427000`
Incremental Recon Attribute	Attribute that holds the date on which the token record was modified.
Object Type	This attribute holds the name of the object type for the reconciliation run. Default value: `User` Do *not* change the default value.
IT Resource Name	Enter the name of the IT resource for the target system installation from which you want to reconcile user records. Default value: `Box`
Resource Object Name	This attribute holds the name of the resource object used for reconciliation. Default value: `Box User` Do *not* change the default value.

3.2.3.2 Box Update Access Token Job

Access token configured as part of IT resource will expire in 60 minutes and refresh token will expire in 60 days. Box Update Access Token Job is used to keep the value of the access token (in the IT resource) always valid. Every 50 minutes, this job is scheduled to run periodically.

> **Note:**
>
> If for some reason this scheduler is not run for more than 60 days, then the refresh token value in IT resource would have expired due to which if you run the Box Update Access Token Job after 60 days, it will fail. In such cases, a new access token and refresh token has to be generated manually.

Table 3-3 describes the attributes of this scheduled job.

Table 3-3 Attributes of the Box Update Access Token Job Schedule Job

Attribute	Description
Access Token Endpoint	This attribute holds the Box REST endpoint to get the new access token. Default value: `https://app.box.com/api/oauth2/token`
IT Resource Name	Enter the name of the IT resource for the target system installation from which you want to reconcile user records. Default value: `Box`
Task Name	This attribute holds the name of the scheduled task. Default value: `Box Update Access Token` You must *not* change the default value.

3.3 Configuring Scheduled Jobs

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

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

To configure a scheduled job:

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

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

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

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

> **Note:**
>
> - 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.
> - See Reconciliation Scheduled Jobs for Box Connector for the list of scheduled tasks and their attributes.

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

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

3.4 Guidelines on Performing Provisioning Operations

These guidelines provide information on what to do when performing provisioning operations.

The following is a guideline that you must apply while performing a provisioning operation:

For a Create User provisioning operation, you must specify a value for the Name and Login fields in the form.

For example:

Name : `John Doe`

Login : `johndoe@example.com`

3.5 Performing Provisioning Operations

You create a new user in Oracle 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 Manager:

1. Log in to Oracle Identity Administrative and User console.
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 created in Step 3, and then click **Checkout.**
5. Specify value for fields in the application form and then click **Ready to Submit**.
6. Click **Submit**.
7. If you want to provision entitlements, then:

   a. On the Entitlements tab, click **Request Entitlements**.

   b. In the Catalog page, search for and add to cart the entitlement, and then click **Checkout**.

   c. Click **Submit**.

3.6 Uninstalling the Box Connector

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

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

4

Extending the Functionality of the Box Connector

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

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

4.1 Adding New User Attributes for Reconciliation

The connector provides a default set of attribute mappings for reconciliation between Oracle Identity Manager and the target system. If required, you can add new user or group attributes for reconciliation.

By default, the attributes listed in Table 1-8 are mapped for reconciliation between Oracle Identity Manager and the target system.

> **Note:**
>
> - This connector supports configuration of already existing (standard) attributes of Box for reconciliation.
> - Only single-valued attributes can be mapped for reconciliation.

The following topics discuss the procedure to add new attributes for users:

- Adding Attributes to the Resource Object
- Creating Reconciliation Field Mapping
- Creating Entries in Lookup Definitions
- Performing Changes in a New UI Form

4.1.1 Adding New Attributes on the Process Form

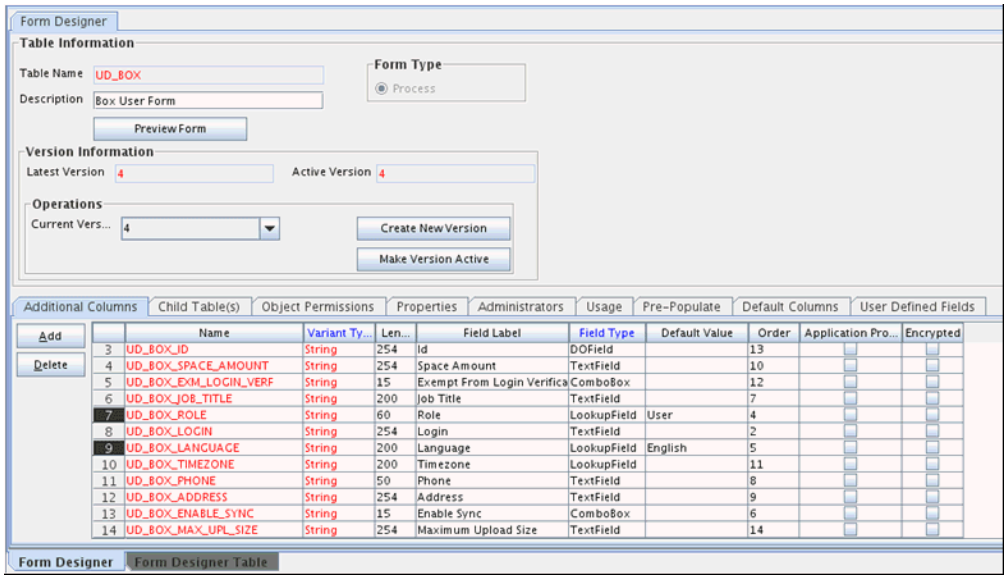
You can add a new attribute on the process form in the Form Designer section of Oracle Identity Manager Design Console.

To add a new attribute on the process form:

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Development Tools**, and double-click **Form Designer**.
3. Search for and open the **UD_Box** process form for users.
4. Click **Create New Version**, and then **Add**.
5. Enter the details of the field.

   For example, if you are adding the **Maximum Upload Size** field, enter `UD_BOX_MAX_UPL_SIZE` in the Name field and then enter other details such as Variable Type, Length, Field Label, and Field Type.
6. Click the Save icon, and then click **Make Version Active**. The following screenshot shows the new field added to the process form.

Figure 4-1 Adding a New Field on the Process Form



4.1.2 Adding Attributes to the Resource Object

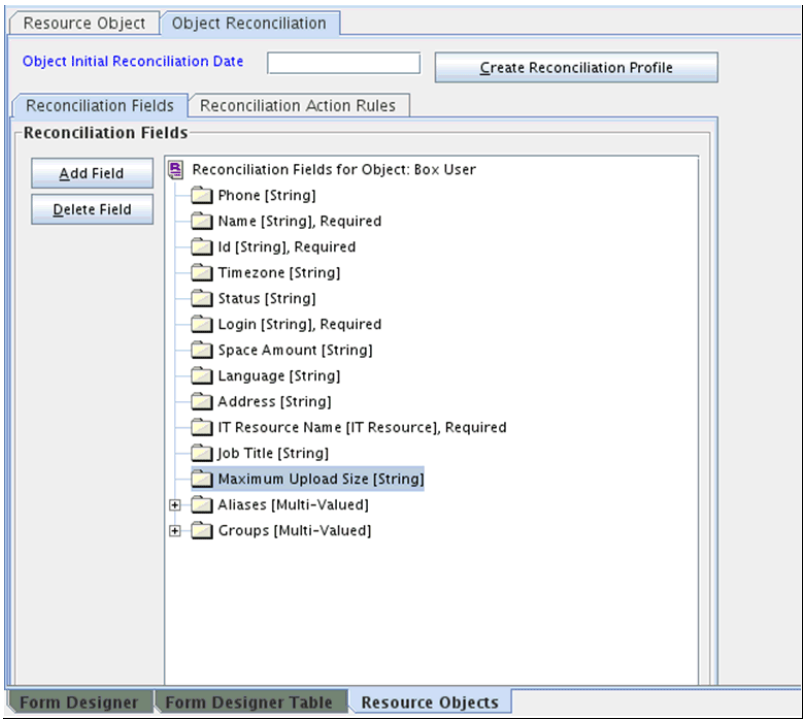
You can add the new attribute to the resource object in the Resource Objects section of Oracle Identity Manager Design Console.

To add the new attribute to the list of reconciliation fields in the resource object:

1. Expand **Resource Management**, and double-click **Resource Objects**.
2. Search for and open the **Box User** resource object for users.
3. On the Object Reconciliation tab, click **Add Field**.
4. Enter the details of the field.

   For example, enter `Maximum Upload Size` in the Name field and select **String** from the Field Type list. Later in this procedure, you enter the field name as the Code value of the entry that you create in the lookup definition for reconciliation.
5. Click the Save icon. The following screenshot shows the new reconciliation field added to the resource object:

Figure 4-2 Newly Added Reconciliation Field



6. Click **Create Reconciliation Profile**. This copies changes made to the resource object into the MDS.

4.1.3 Creating Reconciliation Field Mapping

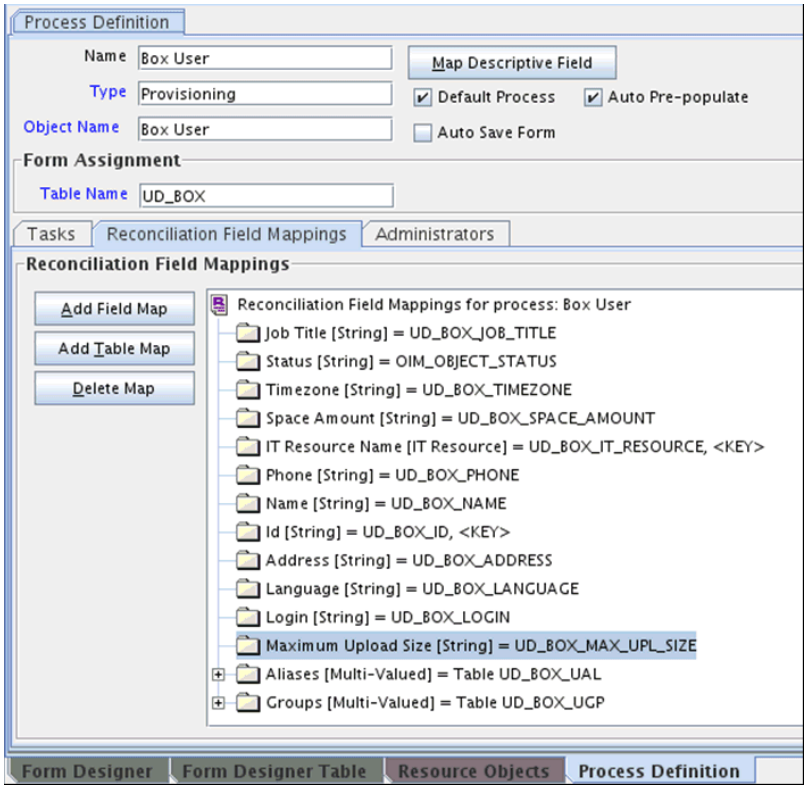
You can create reconciliation field mapping for the new attribute in the Process Definition section of Oracle Identity Manager Design Console.

To create a reconciliation field mapping for the new attribute in the process definition:

1. Expand **Process Management** and double-click **Process Definition**.
2. Search for and open the **Box User** process definition.
3. On the Reconciliation Field Mappings tab of the process definition, click **Add Field Map**.
4. From the Field Name list, select the field that you want to map.

5. Double-click the **Process Data Field** field and select the column for the attribute. For example, select **UD_BOX_MAX_UPL_SIZE**.
6. Click the Save icon. The following screenshot shows the new reconciliation field mapped to a process data field in the process definition:

Figure 4-3 New Reconciliation Field Mapped to a Process Data Field



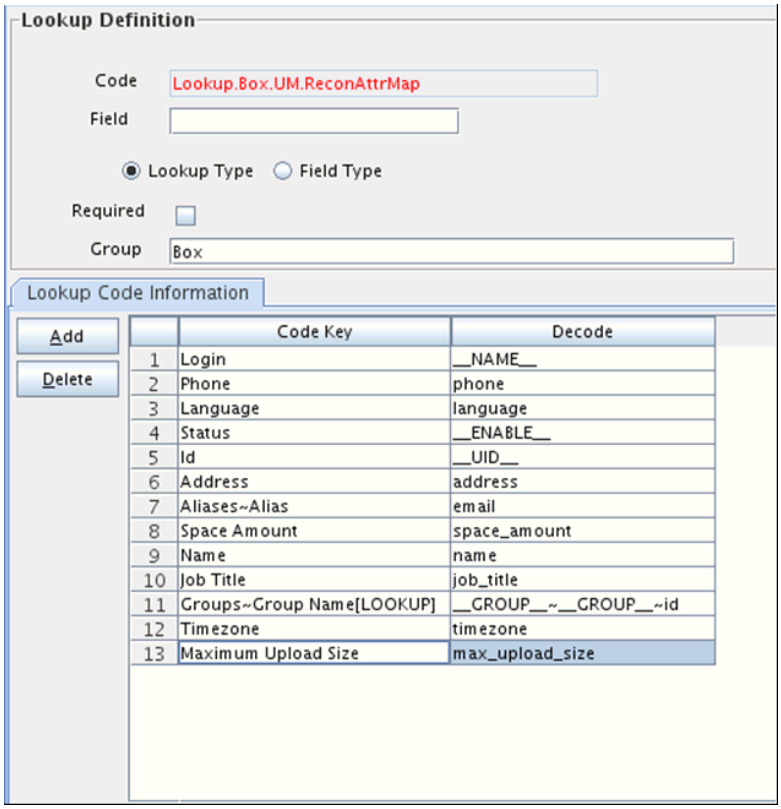
4.1.4 Creating Entries in Lookup Definitions

You must create an entry for the newly added attribute in the lookup definition that holds attribute mappings for reconciliation.

To create an entry for the newly added attribute in the lookup definition:

1. Expand **Administration**.
2. Double-click **Lookup Definition**.
3. Search for and open the **Lookup.Box.UM.ReconAttrMap** lookup definition for users.
4. Click **Add** and enter the Code Key and Decode values for the field. The Code Key value must be the name of the field in the resource object.
5. Click the Save icon. The following screenshot shows the entry added to the lookup definition:

Figure 4-4 Newly Added Entry to Lookup Definition



4.1.5 Performing Changes in a New UI Form

You must replicate all changes made to the Form Designer of the Design Console in a new UI form.

1. Log in to Oracle Identity System Administration.
2. Create and activate a sandbox. See Creating a Sandbox and Activating and Deactivating a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
3. Create a new UI form to view the newly added field along with the rest of the fields. See Creating Forms By Using the Form Designer in *Oracle Fusion Middleware Administering Oracle Identity Manager.*
4. 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, and then save the application instance.
5. Publish the sandbox. See Publishing a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

4.2 Adding New User Attributes for Provisioning

The connector provides a default set of attribute mappings for provisioning between Oracle Identity Manager and the target system. If required, you can add new user or group attributes for provisioning.

By default, the attributes listed in Table 1-11 are mapped for provisioning between Oracle Identity Manager and the target system. If required, you can map additional user attributes for provisioning.

The following topics discuss the procedure to add new user or group attributes for provisioning:

- Adding New Attributes for Provisioning
- Creating Entries in Lookup Definitions for Provisioning
- Creating a Task to Enable Update Operations
- Replicating Form Designer Changes to a New UI Form

4.2.1 Adding New Attributes for Provisioning

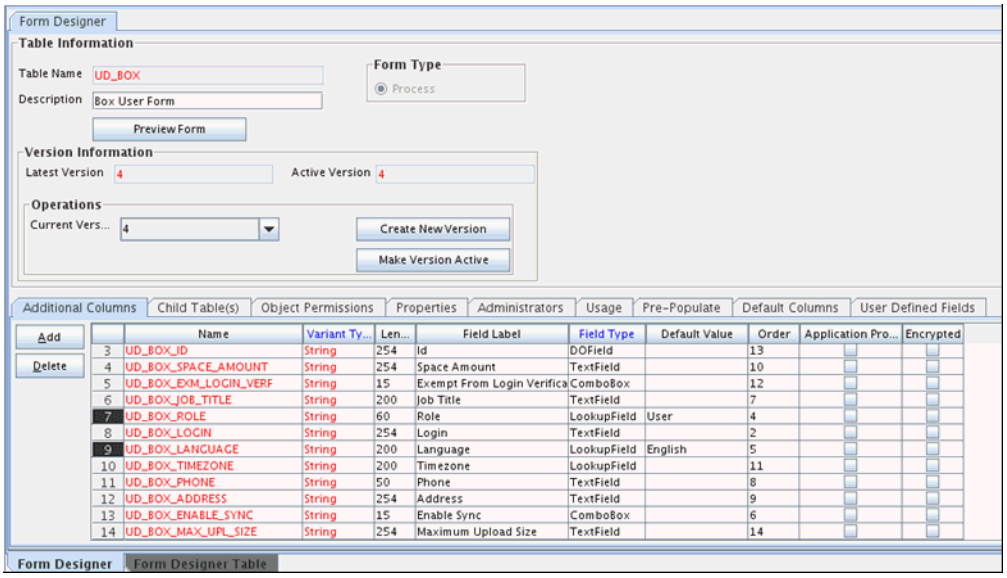
To add a new attribute on the process form, perform the following procedure:

> **Note:**
>
> If you have already added an attribute for reconciliation, then you need not repeat steps performed as part of that procedure.

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Development Tools** and double-click **Form Designer**.
3. Search for and open one the **UD_BOX** process form.
4. Click **Create New Version**, and then click **Add**.
5. Enter the details of the attribute.

   For example, if you are adding the **Maximum Upload Size** field, enter `UD_BOX_MAX_UPL_SIZE` in the Name field, and then enter the rest of the details of this field.
6. Click the Save icon, and then click **Make Version Active**.

Figure 4-5 Newly Added Field



4.2.2 Creating Entries in Lookup Definitions for Provisioning

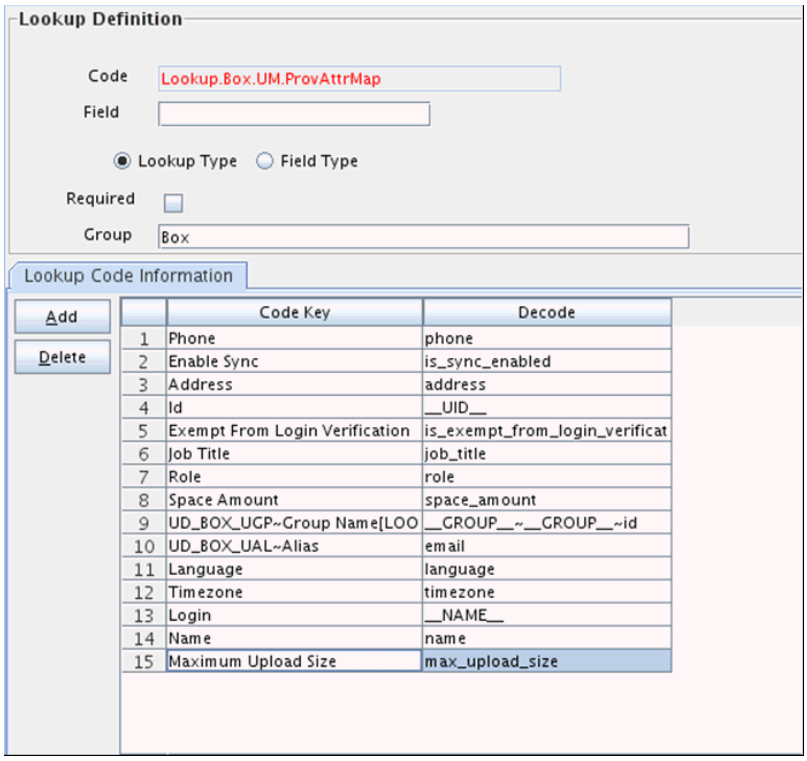
You must create an entry for the newly added attribute in the lookup definition that holds attribute mappings for provisioning.

To create an entry for the newly added attribute in the lookup definition that holds attribute mappings for provisioning:

1. Expand **Administration**.
2. Double-click **Lookup Definition**.
3. Search for and open the **Lookup.Box.UM.ProvAttrMap** the lookup definition.
4. Click **Add** and then enter the Code Key and Decode values for the attribute.

   For example, enter `Maximum Upload Size` in the Code Key column and then enter `max_upload_size` in the Decode column. The following screenshot shows the entry added to the lookup definition:

Figure 4-6 Newly Added Entry to the Lookup Definition



4.2.3 Creating a Task to Enable Update Operations

Create a task to enable updates on the new user or group attribute during provisioning operations.

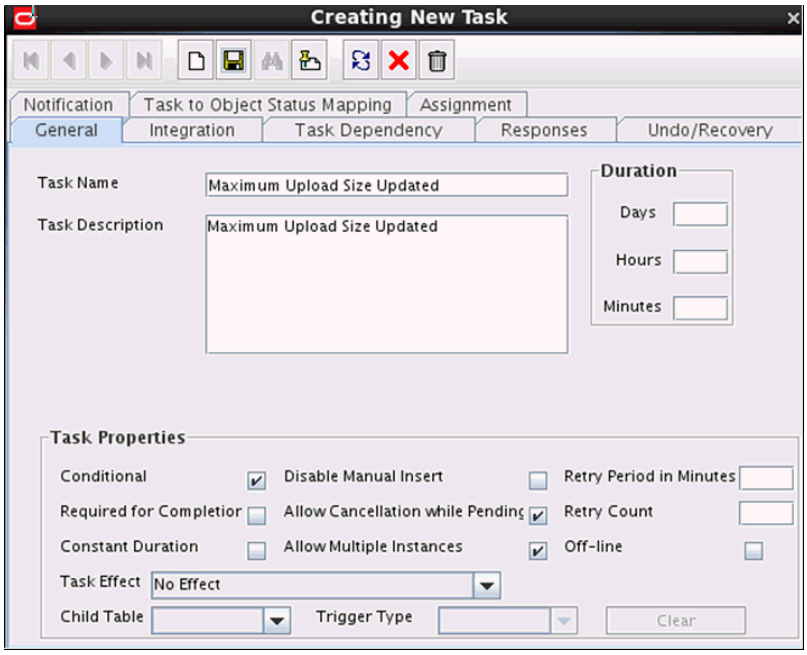
If you do not perform this procedure, then you will not be able to modify the value of the attribute after you set a value for it during the Create User provisioning operation.

To enable the update of the attribute during provisioning operations, add a process task for updating the new user or group attribute as follows:

1. Expand **Process Management** and double-click **Process Definition**.
2. Search for and open the **Box User** process definition.
3. Click **Add.**
4. On the General tab of the Creating New Task dialog box, enter a name and description for the task and then select the following:
   - Conditional
   - Allow Cancellation while Pending
   - Allow Multiple Instances
5. Click the Save icon.

   The following screenshot shows the new task added to the process definition:

Figure 4-7 Newly Added Task to the Process Definition



6. In the provisioning process, select the adapter name in the Handler Type section as follows:

   a. Go to the Integration tab, click **Add.**

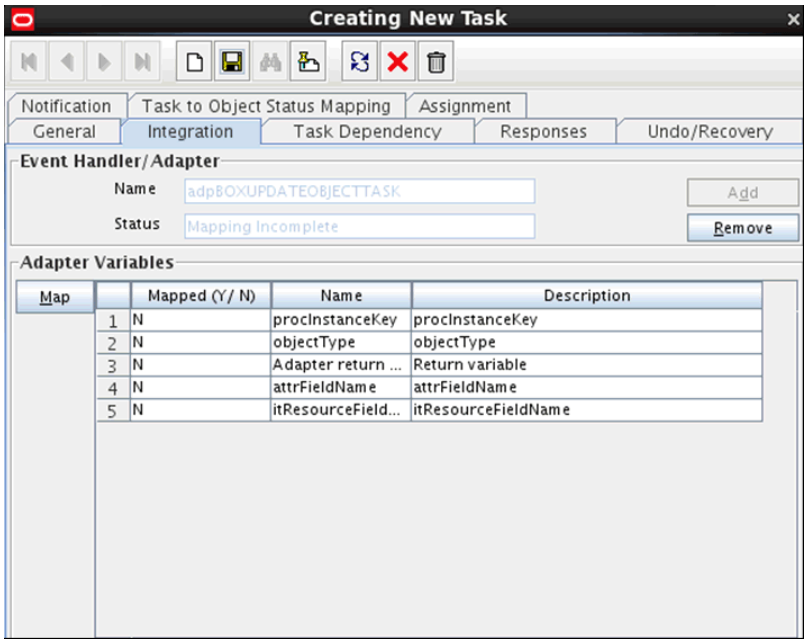
   b. In the Handler Selection dialog box, select **Adapter**.

   c. From the Handler Name column, select **adpBOXUPDATEOBJECTTASK**.

   d. Click Save and close the dialog box.

      The list of adapter variables is displayed on the Integration tab. The following screenshot shows the list of adapter variables:

Figure 4-8 List of Adapter Variables



7. In the Adapter Variables region, click the **procInstanceKey** variable.
8. In the dialog box that is displayed, create the following mapping:
   - **Variable Name:** procInstanceKey
   - **Map To:** Process Data
   - **Qualifier:**Process Instance
9. Click Save and close the dialog box.
10. If you are enabling update provisioning operations for a User attribute, then repeat Steps 7 through 9 for the remaining variables listed in the Adapter Variables region.

    The following table lists values that you must select from the Map To, Qualifier, and Literal Value lists for each variable:

Variable	Map To	Qualifier	Literal Value
Adapter Return Value	Response Code	NA	NA
Object Type	Literal	String	User
itResourceFieldName	Literal	String	UD_BOX_IT_RESOURCE
attributeFieldName	Literal	String	Maximum Upload Size

11. On the Responses tab, click **Add** to add at least the SUCCESS response code, with Status C. This ensures that if the task is successfully run, then the status of the task is displayed as `Completed`.
12. Click the Save icon and close the dialog box, and then save the process definition.

4.2.4 Replicating Form Designer Changes to a New UI Form

You must replicate all changes made to the Form Designer of the Design Console in a new UI form.

1. Log in to Oracle Identity System Administration.
2. Create and activate a sandbox. See Creating a Sandbox and Activating and Deactivating a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
3. Create a new UI form to view the newly added field along with the rest of the fields. See Creating Forms By Using the Form Designer in *Oracle Fusion Middleware Administering Oracle Identity Manager.*
4. 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, and then save the application instance.
5. Publish the sandbox. See Publishing a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

4.3 Configuring Validation of Data During Reconciliation and Provisioning

You can configure validation of reconciled and provisioned single-valued data according to your requirements.

For example, you can validate data fetched from the User Name attribute to ensure that it does not contain the number sign (#). In addition, you can validate data entered in the User Name field on the process form so that the number sign (#) is not sent to the target system during provisioning operations. For data that fails the validation check, the following message is displayed or recorded in the log file: Validation failed for attribute *ATTRIBUTE_NAME*.

> **Note:**
>
> This feature cannot be applied to the Locked/Unlocked status attribute of the target system.

To configure validation of data:

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

   The validation class must implement validate method with the following method signature:

   ```
   boolean validate(HashMap hmUserDetails, HashMap hmEntitlementDetails,
   String field)
   ```

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

```
public boolean validate(HashMap hmUserDetails,
HashMap hmEntitlementDetails, String field) { /*
*    You must write code to validate attributes. Parent
*    data values can be fetched by using hmUserDetails.get(field)
*    For child data values, loop through the
*    ArrayList/Vector fetched by hmEntitlementDetails.get("Child Table")
*    Depending on the outcome of the validation operation,
*    the code must return true or false.
*/
/*
*    In this sample code, the value "false" is returned if the field
*    contains the number sign (#). Otherwise, the value "true" is
*    returned.
*/
               boolean valid=true;
                  String sUserName=(String) hmUserDetails.get(field);
for(int i=0;i<sUserName.length();i++){
if (sUserName.charAt(i) == '#'){ valid=false;
break;}
             }
         return valid;
                }
```

2. Create a JAR file to hold the Java class.
3. Copy the JAR file to Oracle Identity Manager database.

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

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

   - For Microsoft Windows: *`OIM_HOME`*`/server/bin/UploadJars.bat`
   - For UNIX: *`OIM_HOME`*`/server/bin/UploadJars.sh`

   When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Manager administrator, URL of the Oracle Identity Manager host computer, context factory value, type of JAR file being uploaded, and the location from which the JAR file is to be uploaded. Specify 1 as the value of the JAR type.
4. If you created the Java class for validating a process form field for reconciliation, then:
   a. Log in to the Design Console.
   b. Create a lookup definition named **Lookup.Box.UM.ReconValidation.**

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

d. Save the changes to the lookup definition.

e. Search for and open the Lookup.Box.UM.Configuration lookup definition.

f. In the Code Key column, enter `Recon Validation Lookup`. In the Decode column, enter `Lookup.Box.UM.ReconValidation.`

g. Save the changes to the lookup definition.

5. If you have created the Java class for validating a process form field for provisioning, then:

a. Log in to the Design Console.

b. Create a lookup definition by the name Lookup.Box.UM.ProvValidation.

c. In the Code Key column, enter the process form field name. In the Decode column, enter the class name.

d. Save the changes to the lookup definition.

e. Search for and open the Lookup.Box.UM.Configuration lookup definition.

f. In the Code Key column, enter `Provisioning Validation Lookup`. In the Decode column, enter `Lookup.Box.UM.ProvValidation.`

g. Save the changes to the lookup definition.

4.4 Configuring Transformation of Data During User Reconciliation

You can configure transformation of reconciled single-valued account data according to your requirements.

For example, you can use User Name and Last Name values to create a value for the Full Name field in Oracle Identity Manager.

> **Note:**
>
> This feature cannot be applied to the Locked/Unlocked status attribute of the target system.

To configure transformation of single-valued account data fetched during reconciliation:

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

   The transformation class must implement the transform method with the following method signature:

```
Object transform(HashMap hmUserDetails, HashMap hmEntitlementDetails,
String sField)
```

The following sample transformation class creates a value for the Full Name attribute by using values fetched from the User Name and Last Name attributes of the target system:

```
package oracle.iam.connectors.common.transform;
import java.util.HashMap;
public class TransformAttribute {
/*
Description:Abstract method for transforming the attributes
param hmUserDetails< String,Object>
HashMap containing parent data details
param hmEntitlementDetails < String,Object>
HashMap containing child data details
*/
public Object transform(HashMap hmUserDetails, HashMap
hmEntitlementDetails,String sField) {
/*
*    You must write code to transform the attributes. Parent data
attribute values can be fetched by using hmUserDetails.get("Field
Name").
*To fetch child data values, loop through the
*    ArrayList/Vector fetched by hmEntitlementDetails.get("Child Table")
*    Return the transformed attribute.
*/
String sUserName= (String)hmUserDetails.get("User Name"); String
sLastName= (String)hmUserDetails.get("Last Name"); String
sFullName=sUserName+"."+sLastName;
return sFullName;
}
}
```

2. Create a JAR file to hold the Java class.
3. Copy the JAR file to Oracle Identity Manager database.

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

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

   - For Microsoft Windows: *`OIM_HOME`*`/server/bin/UploadJars.bat`
   - For UNIX: *`OIM_HOME`*`/server/bin/UploadJars.sh`

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

4. If you created the Java class for transforming a process form field for reconciliation, then:
    a. Log in to the Design Console.
    b. Create a lookup definition named **Lookup.Box.UM.ReconTransformation**.
    c. In the Code Key column, enter the resource object field name on which you want to apply transformation. For example, User Name. In the Decode column, enter the name of the class that implements the transformation logic. For example, oracle.iam.connectors.common.transform.TransformAttribute.
    d. Save the changes to the lookup definition.
5. Add an entry in the **Lookup.Box.UM.Configuration** lookup definition to enable transformation as follows:
    a. Expand Administration, and then double-click **Lookup Definition**.
    b. Search for and open the **Lookup.Box.UM.Configuration** lookup definition.
    c. In the Code Key column, enter `Recon Transformation Lookup`. In the Decode column, enter `Lookup.Box.UM.ReconTransformation.`
    d. Save the changes to the lookup definition.

4.5 Configuring the Connector for Multiple Installations of the Target System

You must create copies of the connector 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. The company has recently installed Oracle Identity Manager, and they want to configure Oracle Identity Manager to link all the installations of the target system.

To meet the requirement posed by such a scenario, you must create copies of the connector. See Cloning Connectors in *Oracle Fusion Middleware Administering Oracle Identity Manager*.

4.6 Defining the Connector

You can define a customized or reconfigured connector using the Oracle Identity System Administration. Defining a connector is equivalent to registering the connector with Oracle Identity Manager.

A connector is automatically defined when you install it using the Install Connectors feature or when you upgrade it using the Upgrade Connectors feature. You must manually define a connector if:

- You import the connector by using the Deployment Manager.
- You customize or reconfigure the connector.
- You upgrade Oracle Identity Manager.

The following events take place when you define a connector:

- A record representing the connector is created in the Oracle Identity Manager database. If this record already exists, then it is updated.
- The status of the newly defined connector is set to Active. In addition, the status of a previously installed release of the same connector automatically is set to Inactive.

See Defining Connectors in *Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed information about the procedure to define connectors.

A

Files and Directories on the Box Connector Installation Media

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

Table A-1 Files and Directories on the Box Connector Installation Media

File in the Installation Media Directory	Description
bundle/ org.identityconnectors.genericrest-1.0.1115.jar	This JAR is the ICF connector bundle.
configuration/Box-CI.xml	This XML file contains configuration information that is used during connector installation.
Files in the resources directory	Each of these resource bundles contains language-specific information that is used by the connector. During connector installation, these resource bundles are copied to the Oracle Identity Manager database. **Note:** A **resource bundle** is a file containing localized versions of the text strings that include GUI element labels and messages.
xml/Box-ConnectorConfig.xml	This XML file contains definitions for the following connector objects: • IT resource definition • Process forms • Process tasks and adapters • Lookup definitions • Resource objects • Process definition • Scheduled tasks • Reconciliation rules
lib	This directory contains the box-update-accesstoken.jar and its dependent jars which are required to run the Box Update Access Token schedule job.

Index