Oracle® Waveset

8.1.1 Connector Reference

Release 8.1.1

E25959-06

January 2014

The Oracle Waveset 8.1.1 Connector Reference provides reference and procedural information to help you install and configure Oracle Waveset 8.1.1 connectors in order to connect to specific resources and to manage accounts on those resources.

ORACLE

Oracle Waveset 8.1.1 Connector Reference, Release 8.1.1

E25959-06

Primary Author: John Spencer

Contributing Authors: Gowri.G.R, Sridhar Machani, Alankrita Prakash

Contents

3 Oracle Waveset Connector for IBM AS400

6 Oracle Waveset Connector for Google Apps

7 Oracle Waveset Connector for PeopleSoft Employee Reconciliation

8 Oracle Waveset Connector for PeopleSoft User Management

9 Oracle Waveset Connector for SAP User Management

10 Oracle Waveset Connector for Siebel User Management

11 Oracle Waveset Connector for SAP User Management Engine

Index

List of Figures

List of Tables

List of Examples

Preface

The *Oracle Waveset 8.1.1 Connector Reference* provides reference and procedural information to help you install and configure Oracle Waveset 8.1.1 connectors in order to connect to resources and to manage accounts on these resources. This Preface includes the following information:

Who Should Use This Book

This book is intended for deployers and administrators who will install and configure Oracle Waveset 8.1.1 connectors in order to connect to specific resources and to manage accounts on those resources. Readers should have the following experience:

- Deployers should have a background in programming and should be comfortable with XML, Java, Emacs and/or IDEs such as Eclipse or NetBeans.
- Administrators are not required to have a programming background but should be skilled in the specific target resource domain.

Before You Read This Book

Before reading this book, you should be familiar with the *Oracle Waveset 8.1.1 Overview* in the following library:

`http://docs.oracle.com/cd/E19225-01/index.html`

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 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.

How This Book Is Organized

- Chapter 1, "Identity Connectors Overview" provides an overview of the Identity Connector Framework (ICF) and the Oracle Waveset 8.1.1 connectors.
- Subsequent chapters describe the specific Oracle Waveset 8.1.1 connectors that are currently available.

Related Books

For additional information, refer to the other books in the Oracle Waveset 8.1.1 library:

http://docs.oracle.com/cd/E19225-01/index.html

Documentation, Support, and Training

See the following web sites for additional resources:

- Oracle Documentation:
  http://www.oracle.com/technetwork/indexes/documentation/
- Oracle Systems Support:
  http://www.oracle.com/us/support/systems/index.html
- Oracle University: http://education.oracle.com
- Oracle offers a range of resources related to Oracle software on the Oracle Technology Network:

  http://www.oracle.com/technetwork/index.html

  Oracle allows you to:

  - Discuss technical problems and solutions on the Discussion Forums:

    http://forums.oracle.com

  - Get hands-on step-by-step tutorials at the Oracle Learning Library:

    http://www.oracle.com/technetwork/tutorials/index.html

  - Download sample code and scripts at Sample Code for Developers and Admins:

    http://www.oracle.com/technetwork/indexes/samplecode/index.html

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	`$`
Bash shell, Korn shell, and Bourne shell for superuser	`#`
C shell	`machine_name%`
C shell for superuser	`machine_name#`

What's New in the Oracle Waveset 8.1.1 Connector Reference

This chapter provides an overview of the changes in this revision, including:

Software Updates

The following are software updates for Oracle Waveset Connectors:

Software Updates for the Active Directory Connector

The following are software updates for this connector:

Software Updates in Release 11.1.1.6.0

The following are software updates in this release:

Group Account Attribute Added to the User Form In earlier releases, the Group account attribute (__GROUPS__) had to be manually added to the user form. From this release onward, the Group account attribute is available in the user form, by default.

Support for Adding Custom Object Classes with More than One Mandatory Attribute From this release onward, you can add custom object classes with more than one mandatory attribute. See Section 2.3.8, "Adding Custom Object Classes" for more information.

Software Updates in Release 11.1.1.5.0

The following is a software update in this release:

Support for New Target Systems From this release onward, the connector for Microsoft Active Directory adds support for the following target systems:

- Microsoft Active Directory installed on Microsoft Windows Server 2012
- Microsoft Active Directory Lightweight Directory Services installed on Microsoft Windows Server 2012

These target systems are mentioned in Section 2.1.3, "Certified Components for the Active Directory Connector."

Software Updates for the Exchange Connector

The following are software updates for this connector:

Resolved Issues in Release 11.1.1.6.0

The following are issues resolved in this release of the connector:

Bug Number	Issue Description
16799960	The attribute form failed to load that resulted in session time out errors.
14766175	The connector had performance issues that resulted in time out errors.
14666883	The connector failed when it obtained the users from Active Directory.
14547323	The connector failed in patch 145769-08.

Documentation-Specific Updates

The following are the documentation-specific updates in this release:

- Documentation-Specific Updates for this Guide
- Documentation-Specific Updates for the Active Directory Connector
- Documentation-Specific Updates for the Exchange Connector
- Documentation-Specific Updates for the SAP User Management Connector
- Documentation-Specific Updates for the PeopleSoft User Management Connector
- Documentation-Specific Updates for the PeopleSoft Employee Reconciliation Connector

Documentation-Specific Updates for this Guide

The revision with part number E25959-01 replaces the previous version of the *Oracle Waveset 8.1.1 Connector Reference* (part number 821-2844).

This revision has a slightly different format, but the information has not changed (except for new information). The chapter titles are also revised to be more descriptive.

Documentation-Specific Updates for the Active Directory Connector

The following are documentation-specific updates for this connector:

- Documentation-Specific Updates in Release 11.1.1.6.0
- Documentation-Specific Updates in Release 11.1.1.5.0

Documentation-Specific Updates in Release 11.1.1.6.0

The following are documentation-specific updates in this release:

- The "Sync Global Catalog Server" and "Sync Domain Controller" rows have been removed from Table 2–2, " Active Directory Connector Resource Configuration Parameters".
- The "Microsoft .NET Framework" row of Table 2–4, " Active Directory Connector Certified Components" has been modified.
- A note has been added in Section 2.4.2.1, "__ACCOUNT__ Object Class for the Active Directory Connector."
- The following sections have been added:
  - Section 2.3.7, "Adding Auxiliary Classes to Users"
  - Section 2.3.2.4, "Delegating Control of Organizational Units and Custom Object Classes"
  - Section 2.3.2.3.3, "Configuring Log File Rotation"
  - Section 2.4.5, "Enforcing Check Password History"
- Section 2.3.2.3, "Installing and Configuring the .NET Connector Server" has been renamed to Section 2.3.2.3, "Installing, Configuring, and Enabling Logging on the .NET Connector Server" and includes information about enabling logging as well.
- Section 2.5, "Troubleshooting the Active Directory Connector" has been modified. The content of this section have been moved to Section 2.3.2.3.2, "Enabling Logging for the Active Directory Connector."

Documentation-Specific Updates in Release 11.1.1.5.0

The following are documentation-specific updates in this release:

- Section 2.1.3, "Certified Components for the Active Directory Connector," is updated for a Microsoft .NET Framework hotfix that prevents memory leaks.
- Section 4.1.1.1, "Domino Connector Deployment Architecture," is updated for the supported Windows desktop machines.
- The "Microsoft .NET Framework" row of Table 2–4, " Active Directory Connector Certified Components" and Table 5–3, " Exchange Connector Certified Components" has been modified.
- The description of the Database attribute in Table 5–5, " Additional __ACCOUNT__ Attributes for the Exchange Connector" has been modified.

Documentation-Specific Updates for the Exchange Connector

The following are documentation-specific updates for this connector:

Documentation-Specific Updates in Release 11.1.1.6.0

The following are documentation-specific updates in this release:

- Section 5.3.1, "Exchange Connector Deployment Architecture" has been updated with new architecture information.
- Section 5.5, "Frequently Asked Questions (FAQs)" has been added.
- Section 5.6, "Troubleshooting Connector Issues" has been added.

Documentation-Specific Updates for the SAP User Management Connector

The following are documentation-specific updates for this connector:

- The "Oracle Waveset" row of Table 9–1, " SAP User Management Connector Operations" has been updated.
- Section 9.2.1, "Downloading and Installing the SAP Java Connector (JCo) Files" has been updated.
- Section 9.5.1, "Multi-valued Attributes Prefixed with Underscore" has been added.

Documentation-Specific Updates for the PeopleSoft User Management Connector

The following is a documentation-specific update for this connector:

- The "Target systems", "People Tools", and "JDK" rows of Table 8–3, " Certified Components for the PeopleSoft User Management Connector" have been updated.

Documentation-Specific Updates for the PeopleSoft Employee Reconciliation Connector

The following are documentation-specific updates for this connector:

- The "Target systems", "People Tools", and "JDK" rows of Table 7–2, " Certified Components for the PeopleSoft Employee Reconciliation Connector" have been updated.
- A note has been added to the following sections:
  - Section 7.3.1.2.2, "Records"
  - Section 7.3.1.2.5, "Component Interfaces"
- Section 7.3.1.7, "Step 7: Testing Component Interface" has been added.
- A new file has been added to the list of file names in Step 3 of Section 7.3.4, "Postinstallation Tasks for the PeopleSoft Employee Reconciliation Connector."

1 Identity Connectors Overview

An identity connector is a component, similar to a resource adapter, that provisions and reconciles users to a target resource, such as IBM Domino or AS400. The scope of identity connectors has expanded to support both Oracle Identity Manager (OIM) and Oracle Waveset. That is, a connector can be installed on either system, although the connectors perform provisioning and reconciliations in different ways.

This document describes connectors from the Oracle Waveset perspective. Oracle Identity Manager provides a separate publication for each connector. Refer to the OIM publications for detailed information about each connector from the Oracle Identity Manager perspective.

The following sections describe connectors and the connector architecture:

1.1 Connector Architecture Overview

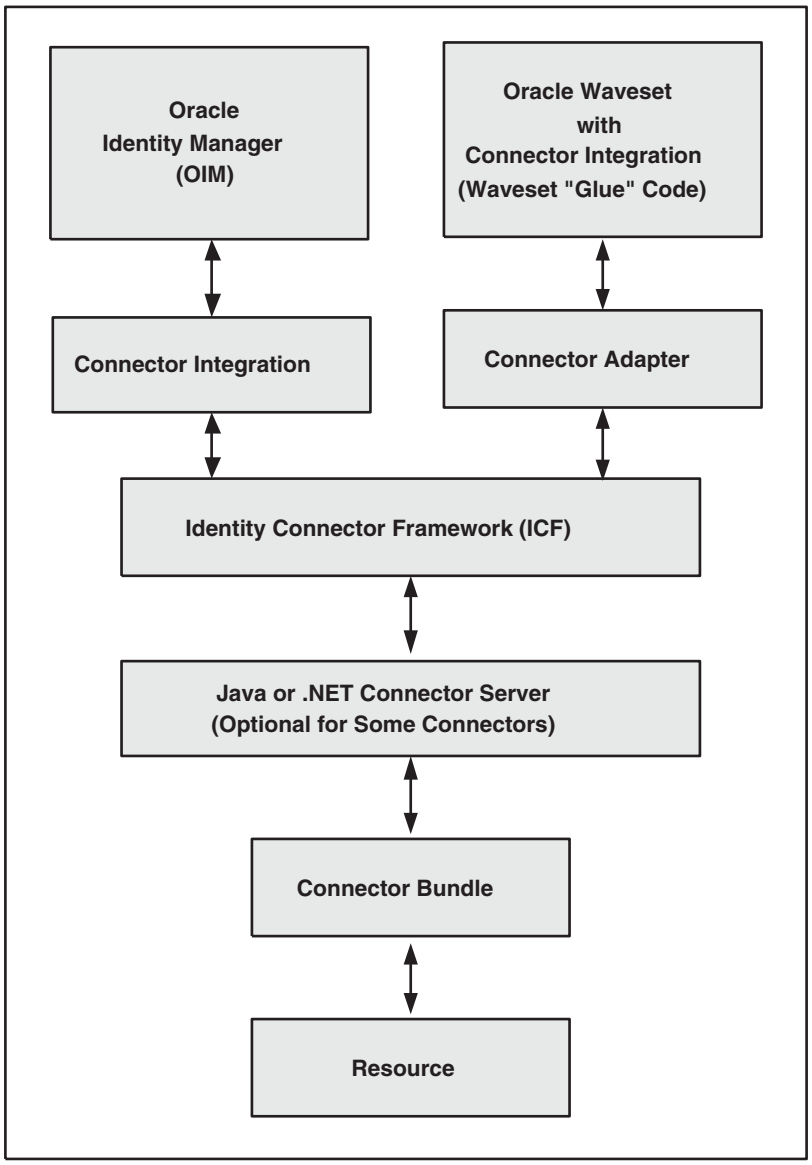
Sun Identity Manager first supported connectors in version 8.1. Connectors were introduced because they provided the following advantages over resource adapters:

- Connector release cycles do not rely upon Oracle Waveset release cycles. As a result, you can add or update connectors in your deployment with less dependence on the specific version of Oracle Waveset you are currently using.
- Oracle Waveset loads each connector in a separate class loader. This enhances support for using multiple versions of a native API from within a single Oracle Waveset server.

Oracle Waveset connectors are designed to appear nearly identical to resource adapters in the administrator interface. However, the underlying architecture is very different.

The following figure shows the identity connector architecture.

Figure 1–1 Identity Connector Architecture



Each of these components are described in the following sections:

- Identity Connector Framework
- Connector Bundles
- Oracle Waveset Connector Integration Files
- Connector Servers

1.1.1 Identity Connector Framework

The Identity Connector Framework (ICF) is a component that provides basic provisioning, reconciliation, and other functions that all Oracle Identity Manager and Oracle Waveset connectors require.

Oracle Waveset includes the ICF in the `WEB-INF` directory. You do not need to configure or modify the ICF.

The ICF is packaged in the following JAR files.

- ICF Application Programming Interface (API): `connector-framework.jar`
- ICF implementation: `connector-framework-internal.jar`
- Embedded Groovy script interpreter: `groovy-all.jar`

The ICF provides a Service Provider Interface (SPI) for creating a custom connector or extending an existing connector. The use of this SPI is not described in this document.

1.1.2 Connector Bundles

A connector bundle is simply a specialized JAR file that contains the connector implementation. The connector bundle can also include any auxiliary libraries such as third-party libraries that are required by a connector.

If Oracle has a license to distribute a resource's corresponding JAR file, it is also included in the bundle. However, in most cases, you will need to add the JAR to the bundle or put the JAR into the `WEB-INF/lib` directory of Oracle Waveset server. The documentation for each connector indicates whether this is necessary and provides information about where to place external JAR files. The directory for third-party JAR files is the `/lib` subdirectory of the connector bundle.

A bundle includes a manifest file, `META-INF/MANIFEST.MF`, which defines the following properties:

- `ConnectorBundle-FrameworkVersion` — The minimum version of the ICF required by the connector. This requirement is checked against version of deployed ICF inside Oracle Waveset.
- `ConnectorBundle-Name` — The qualified name for the connector bundle, such as `org.identityconnectors.as400`.
- `ConnectorBundle-Version` — The version of the bundle. Within a given deployment, the combination of `ConnectorBundle-Name` and `ConnectorBundle-Version` must be unique.

You can implement multiple connectors that each support a different version of a resource by using different versions of the connector bundle. In this case, it is highly recommended that you place the resource's JAR file inside the connector bundle.

The Oracle Waveset 8.1.1 bundle patch automatically installs the bundle files. If you need to deploy the bundle in a WAR file, place the bundle in the `WEB-INF/bundles` directory at the root of the web application.

1.1.3 Oracle Waveset Connector Integration Files

The Oracle Waveset connector integration files are XML files that provide the configuration information necessary to transform data from a resource to Oracle Waveset. These integration files are sometimes called the connector "glue" code. If you are upgrading from a resource adapter to the current version of the connector, consider reviewing the respective XML files before you do the upgrade.

Each Oracle Waveset connector has specific integration files that are distributed as part of the connector bundle. These files are in a directory that includes the name `glue`.

Although the specific contents of the connector integration files vary with each connector, these files typically contain the following information:

- Version of the code
- Schema map

- Resource attributes
- Migration instructions, including adding and deleting account and resource attributes
- Definitions of resource-specific objects
- Custom user forms

1.1.4 Connector Servers

A connector server is required when a connector bundle is not directly executed within an application. By using one or more connector servers, the connector architecture allows an application to communicate with externally deployed bundles. The following connector servers are available, depending on the platform:

- Java Connector Server

  The Java Connector Server is available on Windows, Linux, and UNIX systems. The Java Connector Server is useful when you do not want to execute a Java connector bundle in the same virtual machine (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. Additionally, you can use the Java Connector Server in order to eliminate the possibility of an application VM crash due to a fault in a JNI-based connector.

  Some connectors such as the Domino connector require the Java Connector Server, but for other connectors, the Java Connector Server is optional.

- .NET Connector Server

  The .NET connector server is available for the Microsoft .NET Framework on Windows systems.

1.2 Installing Components

The ICF and connector bundles are installed automatically when you install the appropriate Oracle Waveset 8.1.1 bundle patch. The connectors might require additional installation steps and configuration. Refer to the connector-specific documentation for information about installing and configuring a connector.

Because connector servers reside on a machine other than the application server, they must be installed manually. Refer to the connector-specific documentation for information about installing and configuring a connector server.

1.3 Additional Management Topics

The following sections describe the following connector-related management tasks in an Oracle Waveset deployment:

- Changing the Connector Server or Version Used by a Resource
- Setting a Time-Out for a Connector-Based Resource
- Editing Connection Pooling Parameters
- Using Resource Actions with Connector-Based Resources
- Removing a Connector from a Deployment

1.3.1 Changing the Connector Server or Version Used by a Resource

When you create a resource, Oracle Waveset writes information about the selected connector server to the resource object. You can change the connector server of an existing resource, or change the version of the connector.

To change connector server information in the resource object:

1. From the Resource page, select the resource you want to edit.
2. Select the Resource Actions> Change Connector Parameters menu option. Note that Oracle Waveset permits you to select only a connector server that has at least one version of the connector available. The only versions displayed are those provided by the selected connector server.

1.3.2 Setting a Time-Out for a Connector-Based Resource

When you are editing or creating a connector-based resource, Oracle Waveset displays a set of fields known as operation time-outs. By default, Oracle Waveset sets operation time-outs to a value of -1, which represents no time-out. When you set this field to a non-zero value, the operation times out with an error if the connector does not complete the operation sooner than the specified time-out interval. Oracle Waveset stores time out values in the Resource XML object under the `<OperationTimeouts>` tag. Time-outs with a value of -1 are not stored in the XML.

1.3.3 Editing Connection Pooling Parameters

When editing a connector-based resource, you will see the Connector Pooling configuration fields on the last page of the resource wizard. On that page, you can set values for these attributes:

- **Maximum Objects** is the maximum number of connector instances that can exist simultaneously. The number of pooled idle instances and active instances cannot exceed this number.
- **Maximum Idle Objects** is the minimum number of idle connector instances that will be held in the pool.
- **Minimum Idle Evict Time** is the minimum time (milliseconds) to wait before evicting an idle object from the pool when more than the Minimum Idle Objects are already pooled. Zero (0) means to evict the object immediately.
- **Maximum Wait** is the maximum time (milliseconds) to wait for a connector instance to become available when Maximum Objects already exists.

1.3.4 Using Resource Actions with Connector-Based Resources

Connector-based resources follow the same rules as adapter-based resources in terms of defining resources actions to use as before and after actions. Oracle Waveset supports the use of before and after actions, including create, update, delete, disable, and enable operations.

1.3.5 Removing a Connector from a Deployment

You remove a connector from deployment by removing its corresponding JAR or DLL file. Once the connector is removed, Oracle Waveset can no longer access it. If you remove a connector from deployment while Oracle Waveset resources still reference it for their implementation, any further use of that resource within Oracle Waveset will

result in run-time errors. To help prevent this problem, run the Connectors-In-Use report before removing connectors from deployment.

1.4 Debugging and Troubleshooting

Oracle Waveset provides the following types of tracing for connector performance:

- API-Layer Tracing
- Java Connector Framework Tracing
- Java Connector-Specific Tracing
- .NET Tracing

Tracing of local Java connectors can be limited on a class level only. This differs from the method-level tracing supported for other classes. Oracle Waveset does not support the ability to manage tracing on remote connectors.

1.4.1 API-Layer Tracing

Use this level of tracing to determine whether the problem is within Oracle Waveset or the connector itself. This trace method works for both remote and local connectors. To enable connector API-level tracing, enable level 4 Oracle Waveset tracing for class `org.identityconnectors.framework.impl.api.LoggingProxy`. This type of tracing focuses on the arguments and return values of every connector API method call.

1.4.2 Java Connector Framework Tracing

To implement, enable Oracle Waveset tracing for the connector Java classes (for example, `org.identityconnectors.framework.*`). This trace method works with all log calls made internally by the framework implementation classes.

1.4.3 Java Connector-Specific Tracing

Use this level of tracing to troubleshoot problems within a connector. This trace method works only for local Java connectors. To implement, enable Oracle Waveset tracing for the connector Java classes. It traces all log calls made by the connector code into the Oracle Waveset trace file.

1.4.4 .NET Tracing

.NET connectors call the standard .NET trace API. Oracle Waveset does not provide centralized tracing control. You cannot view .NET trace files from within Oracle Waveset. You must edit the local connector server configuration file to configure .NET tracing.

2

Oracle Waveset Connector for Microsoft Active Directory

This chapter includes the following information about the Active Directory connector for Oracle Waveset:

2.1 About the Active Directory Connector

2.1.1 Overview of the Active Directory Connector

The Oracle Waveset Active Directory connector is a .NET connector that supports provisioning to Microsoft Windows servers running:

- Microsoft Active Directory Domain Services (AD DS)
- Microsoft Active Directory Lightweight Directory Services (AD LDS), formerly called Active Directory Application Mode (ADAM)

  For AD LDS, the Active Directory connector supports the same features that are supported for AD DS, except when explicitly noted otherwise in this chapter.

  Oracle does not provide AD LDS specific integration artifacts (that is, glue) for Oracle Waveset. Therefore, to use AD LDS, you must manually configure the Oracle Waveset resource, user forms, and any related artifacts. (Or, you can use the LDAP resource adapter.) For more information, see Configuring the Active Directory Connector for AD LDS.

The Active Directory connector is implemented using the Identity Connector Framework (ICF). The ICF provides a container that separates the connector bundle from the application. The ICF also provides common features such as connection

pooling, buffering, time outs, and filtering to simplify the usage of the connectors. For more information about the ICF, see Chapter 1, "Identity Connectors Overview".

Other considerations for the Active Directory connector are:

- The Active Directory connector operates in the context of the .NET Connector Framework, which in turn requires an application to execute. Therefore, by default, Oracle provides (and recommends) the .NET Connector Server to run the Active Directory connector.
- The Active Directory connector supports any scripting language that has a script executor in the ICF. Currently, there are two script executor implementations: a Windows shell script executor (batch scripts) and a Boo script executor. Although Visual Basic scripts are not directly supported, a Visual Basic script can be called via a shell script.
- The Active Directory connector supports agentless target deployment; that is, an agent is not required.
- The Active Directory connector supports various Active Directory Forest Topologies The connector supports Global Catalog Server based reconciliation to bring the objects from the nested (child) domain.

  The connector also supports reconciliation and provisioning operations across domains. For example, you can assign a user in one domain to a group in another domain. You can also reconcile a user record even if the user and the user's manager belong to different domains. The connector supports provisioning of the accounts in the child domain by connecting to the parent domain and then using referrals for provisioning to the child domain.
- The Active Directory connector supports a custom schema. You can create custom object classes and different object classes for different types of users. The connector supports provisioning and reconciliation for custom object classes.
- The Active Directory connector supports fail-over. The backup domain controller URLs must be configured for handling fail-over. The connector will connect to the available backup domain controller if the primary domain controller is not reachable.
- The Active Directory connector supersedes the Active Directory resource adapter. For migration information, see Migrating an Active Directory Resource Adapter.

This section provides the following additional information about the Active Directory connector:

- Active Directory Connector Features
- Active Directory Connector Resource Configuration Parameters
- Identity Template for the Active Directory Connector

2.1.1.1 Active Directory Connector Features

The Active Directory connector supports the following operations:

Table 2–1 Active Directory Connector Operations

Operation	Description
Account provisioning	■ Create, read, update, and delete objects ■ Enable, disable, and rename accounts ■ User Provides Password On Change option. See User Provides Password On Change for the Active Directory Connector. ■ Pass-through authentication ■ Before and after actions. See Before and After Actions for the Active Directory Connector.
Reconciliation	Full and incremental reconciliation. See also Reconciliation for the Active Directory Connector. Data loading methods include: ■ Import directly from the target system ■ Reconcile with the target system ■ Active Sync. See Active Sync for the Active Directory Connector.

2.1.1.1.1 Reconciliation for the Active Directory Connector Reconciliation compares the contents of the account index to what each resource currently contains. The Active Directory connector supports both full and incremental reconciliation. Reconciliation can perform the following functions:

- Detect new and deleted accounts
- Detect changes in account attribute values
- Correlate accounts with Oracle Waveset users
- Detect accounts that are not associated with Oracle Waveset users
- Detect when a user has been moved from one container on a resource to another container on a resource

2.1.1.1.2 Active Sync for the Active Directory Connector Active Sync listens or polls for changes to a resource, detecting incremental changes in real time. Active Sync must always connect to the same Active Directory server. If the Search Child Domains configuration property is not set, the Sync Domain Controller property must be configured to specify the hostname of a specific Sync Domain Controller because Active Sync must always connect to the same Domain Controller. If the Search Child Domains property is set, then the Sync Global Catalog Server property must be set to a specific Global Catalog server.

For information about limiting the number of repeated events that occur when you switch to a new domain controller, see Chapter 53, "Active Directory Synchronization Failover," in the *Oracle Waveset 8.1.1 Resources Reference* in the following library:

`http://docs.oracle.com/cd/E19225-01/index.html`

If the Active Directory connector is configured to sync from AD LDS, the Active Directory Domain Controller Hostname defines the server to contact. The Search Child Domain configuration property is ignored for AD LDS.

2.1.1.1.3 Before and After Actions for the Active Directory Connector The Active Directory connector supports before and after actions, which use scripts (written in a supported

scripting language) to perform activities on the Connector Server during a user create, update, or delete request.

For more information, see Chapter 51, "Adding Actions to Resources," in the *Oracle Waveset 8.1.1 Resources Reference* in the following library:

`http://docs.oracle.com/cd/E19225-01/index.html`

Before and after actions are mapped to the `ScriptOnConnectorOp` and `ScriptOnResourceOp` SPI operations in the ICF. The `ScriptOnConnectorOp` operation is supported by the default implementation of the framework.

See also Configuring Before and After Actions for the Active Directory Connector.

2.1.1.1.4 User Provides Password On Change for the Active Directory Connector When a user's password is changed, to meet the password history requirements, the user might need to provide the previous password. The "User Provides Password On Change" option (WS_USER_PASSWORD attribute) is available in the Active Directory connector userForm.xml file. Several considerations are:

- This attribute is ignored if the "PasswordNever Expires" option is set for the resource.
- This attribute is not available in the Active Directory connector resource configuration page.

2.1.1.1.5 Support for Failover of Active Directory Target Systems The BCDHostNames parameter provides support for failover of replicated target systems. If a target system goes down, operations can still be performed using a backup domain controller host. The target systems will have specific recovery methods (independent of the Active Directory connector), which will support the data synchronization for the replication of the terminals and hosts. The requirements for failover support include:

- The target systems must be in the same domain.
- The target systems must be SSO enabled (or have the same authorization credentials).
- The Active Directory hierarchy must be the same (true copy).

For more information, see the BCDHostNames parameter in Table 2–2, " Active Directory Connector Resource Configuration Parameters".

2.1.1.2 Active Directory Connector Resource Configuration Parameters

The following table describes the configuration parameters that you specify when you configure a resource for the Windows target system. The resource contains connection information about the target system.

Table 2–2 Active Directory Connector Resource Configuration Parameters

Parameter Name	Type	Required	Description
Directory Administrator's Account	String	Yes	Administrator's user name with which the system should authenticate. Can be either a username or a combination of domain name and user name in the form of `'domainname'\'username'`. For example: Administrator
Directory Administrator's Password	String	Yes	Administrator's password that should be used when authenticating.

Table 2–2 (Cont.) Active Directory Connector Resource Configuration Parameters

Parameter Name	Type	Required	Description
Object Class for User Objects	String	No	Active Directory object class for user objects that will be managed on the specified resource. The default is User (which for most situations should be sufficient).
Container	String	Yes	Base context for all searches. For example: `OU=finance,DC=example,DC=com`
Create Home Directory	Boolean	No	Specifies whether or not the home directory for the user will be created.
Active Directory Domain Controller Hostname	String	No	Domain controller: hostname, IP address, or domain name of the LDAP server. If not supplied, a serverless bind is used. For example: `10.0.0.1` If the Active Directory connector is configured to sync from AD LDS, this parameter defines the server to contact.
Search Child Domains	Boolean	No	Set if you want searches of Active Directory to include child domains. In addition, the Search Container and Sync Search Context (see the sync settings) attributes must be set to the top of the parent domain. For example: `DC=mydomain,DC=com` **Note**. The parameter is ignored for AD LDS.
Domain Name	String	Yes	Name of the Windows domain. For example: `finance.example.com`
Search Context	String	No	Not currently used.
Use SSL	Boolean	No	Select if the connection to the target system must be encrypted through an SSL channel. The default is No. If set to Yes, this parameter enables SSL between the .NET Connector Server where the Active Directory bundle is deployed and Active Directory or AD LDS. **Note**. Even if the value is set to No, communication between the .NET Connector Server Active Directory will be of "Secure type".
Delete Leaf Nodes of User Objects	Boolean	No	Select if the associated leaf nodes of a User object to delete are intended to be removed along with the object. If not selected and the User object to delete has leaf nodes, the operation will fail and an error message will be displayed.

Table 2–2 (Cont.) Active Directory Connector Resource Configuration Parameters

Parameter Name	Type	Required	Description
Page Size	Integer	No	Indicates the page size returned from Active Directory queries to the connector in a paged search. The default is 1000. Paging splits the entire result set of a query into smaller subsets called, appropriately, pages. In general, it is recommended that you set this value to the maximum page size for simple searches. By setting the parameter to the maximum value, you can minimize the network round trips necessary to retrieve each page, which tends to be the more expensive operation for simple searches. While you can specify a Page Size value greater than the Active Directory system's MaxPageSize value, the Active Directory server will ignore the Page Size value and use the MaxPageSize value instead. No exception will be generated in this case. In some cases, you might need to specify a smaller Page Size value to avoid time outs or overtaxing the server. Some queries are especially expensive, so limiting the number of results in a single page can help avoid this problem.
Lockout Threshold in AD LDS	Integer	No	Specifies the configured number of failed logon attempts that causes a user account to be locked out in an AD LDS instance.
AD LDS Port	Integer	No	Specifies the port number on which the AD LDS instance is listening for connections.
Target is an AD LDS (ADAM) instance	Boolean	No	Select if the target system is an AD LDS instance. The default is No.
BCDHostNames	String	No	To set BCDHostNames, specify the Active Directory domain controller host names separated by a semicolon. For example: `host1.domain.com;host2.domain.com`

2.1.1.3 Identity Template for the Active Directory Connector

Windows Active Directory is a hierarchically based resource. The Active Directory connector identity template provides the default location in the directory tree where a user is created. The default identity template is:

```
CN=$fullname$,cn=Users,dc=mydomain,dc=com
```

Note: For a container name, you must specify ou, cn, and dc in lower case.

You must replace the default template with a valid template for your deployment.

2.1.2 Security Considerations for the Active Directory Connector

2.1.2.1 Secure Communication to the Target System

On the Active Directory connector side, secure communication is ensured by the API. Any bind to the directory is secured by the Windows Security Support Provider Interface (SSPI). Because password management requires an SSL channel for AD LDS, the Active Directory connector can be configured to communicate with the target system via an SSL channel.

The communication between the .NET Connector Framework and Oracle Waveset is encrypted by the framework, but it is also recommended that you use an SSL connection.

2.1.2.2 Active Directory Administrative Account Permissions

The administrative account configured in the Active Directory resource must have the permissions in Active Directory as shown in the following table.

Table 2–3 Active Directory Administrative Account Permissions

Oracle Waveset Functionality	Active Directory Administrative Account Permissions
Create user	Create User Objects To create the account enabled, you must have the ability to Read/Write the `userAccountControl` property. To create with the password expired, you must be able to Read/Write the `Account Restrictions` property set (includes the `userAccountControl` property).
Delete user	Delete User Objects
Update users	Read All Properties, Write All Properties **Note**: If only a subset of the properties are to be managed from Waveset, then Read/Write access can be given to just those properties.
Change/Reset passwords Unlock user accounts Expire Active Directory user accounts	User Object permissions: ■ List Contents ■ Read All Properties ■ Read Permissions ■ Change Password ■ Reset Password User Property permissions: ■ Read/Write `lockoutTime` Property ■ Read/Write `Account Restrictions` Property ■ Read `accountExpires` Property

2.1.2.3 .NET Connector Server Service Account Considerations

By default, the .NET Connector Server runs as the local system account. This option is configurable through the Services MMC Snap-in.

if you run the .NET Connector Server as an account other than Local System, the Connector Server service account requires the "Act As Operating System" and "Bypass Traverse Checking" user rights. It uses these rights for pass-through authentication and for changing and resetting passwords in certain situations.

Most of the management of Active Directory is done using the administrative account specified in the resource. However, some operations are done as the Connector Server service account. Thus, the Connector Server service account must have the appropriate permissions to perform these operations. Currently, these operations are:

- Creating home directories
- Running actions (including before and after actions)

When performing before and after action scripts, the .NET Connector Server might need the "Replace a process level token" right. For example, this right is required if the .NET Connector Server attempts to run the script subprocess as another user, such as the resource administrative user. in this case, the .NET Connector Server process needs the right to replace the default token associated with that subprocess.

if this right is missing, the following error can be returned during subprocess creation:

```
"Error creating process: A required privilege is not held by the client"
```

The "Replace a process level token" right is defined in the Default Domain Controller Group Policy object and in the local security policy of workstations and servers. To set this right on a system, open the Local Security Policies application within the Administrative Tools folder, then navigate to Local Policies, User Rights Assignment, and then Replace a process level token.

2.1.3 Certified Components for the Active Directory Connector

The Active Directory connector is certified with the following components:

Table 2–4 Active Directory Connector Certified Components

Component	Requirement
Oracle Waveset	Oracle Waveset 8.1 Update 1 Bundle Patch 6 or later
Identity Connector Framework (ICF)	ICF 1.2 or later

Table 2–4 (Cont.) Active Directory Connector Certified Components

Component	Requirement
Microsoft .NET Framework	Microsoft .NET Framework 3.5 or later **Note:** To prevent a memory leak problem with Microsoft .NET Framework 3.5, apply the hotfix described in the following article: http://support.microsoft.com/kb/981575
Target Systems	**Microsoft Active Directory** ■ Microsoft Active Directory installed on Microsoft Windows Server 2003, both 32-bit and 64-bit platforms ■ Microsoft Active Directory installed on Microsoft Windows Server 2003 R2, both 32-bit and 64-bit platforms ■ Microsoft Active Directory installed on Microsoft Windows Server 2008, both 32-bit and 64-bit platforms ■ Microsoft Active Directory installed on Microsoft Windows Server 2008 R2, both 32-bit and 64-bit platforms ■ Microsoft Active Directory installed on Microsoft Windows Server 2012, 64-bit platform
Target Systems (continued)	**MicrosoftActive Directory Lightweight Directory Services (AD LDS) or MicrosoftActive Directory ApplicationMode (ADAM)** ■ Microsoft ADAM installed on Microsoft Windows Server 2003, both 32-bit and 64-bit platforms ■ Microsoft ADAM installed on Microsoft Windows Server 2003 R2, both 32-bit and 64-bit platforms ■ Microsoft AD LDS installed on Microsoft Windows Server 2008, both 32-bit and 64-bit platforms ■ Microsoft AD LDS installed on Microsoft Windows Server 2008 R2, both 32-bit and 64-bit platforms ■ Microsoft AD LDS installed on Microsoft Windows Server 2012

2.1.4 Supported Languages for the Active Directory Connector

The Active Directory connector is localized in the following languages:

- Arabic
- Chinese (Simplified and Traditional)
- Danish
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazilian)
- Spanish

2.2 Migrating an Active Directory Resource Adapter

> **Note:** The migration XML covers most of the cases. However, if your resource is customized so that the default migration is not applicable, you must edit the XML file for the `MigrationForm.Org.IdentityConnector.ActiveDirectory.ActiveDirectoryConnector` form before starting the migration process.

Before You Get Started: Install and configure the latest version of the .NET Connector Server, as described in Installing, Configuring, and Enabling Logging on the .NET Connector Server.

To migrate a Active Directory resource adapter to the Active Directory connector, follow these steps:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the Active Directory Connector.
2. Log in to the Oracle Waveset Administrator interface.
3. Go to the Migrate Adapters page.
4. Select the Resource you want to migrate from.
5. Select the type of the connector you want to convert to.
6. On the next page, select the Active Directory connector version and the Connector Server to use.
7. Provide the Active Directory Domain Name.
8. Click Convert.

> **Note:** After you finish the migration, it is recommended that you test the configuration.

2.3 Deploying the Active Directory Connector

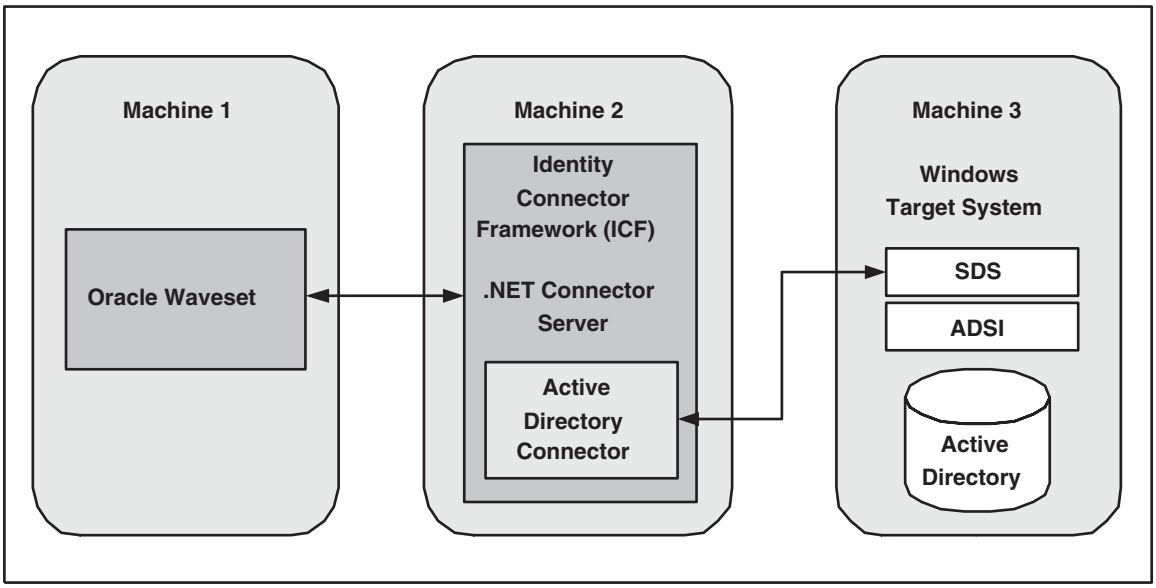
This section discusses the following topics:

- Active Directory Connector Deployment Architecture With the .NET Connector Server
- Preinstallation Tasks for the Active Directory Connector
- Installing the Active Directory Connector
- Postinstallation Tasks for the Active Directory Connector
- Configuring the Active Directory Connector for AD LDS
- Enabling Reconciliation and Provisioning Operations Across Multiple Domains
- Adding Auxiliary Classes to Users
- Adding Custom Object Classes

2.3.1 Active Directory Connector Deployment Architecture With the .NET Connector Server

The following figure shows the distributed deployment architecture with the Active Directory connector deployed in the .NET Connector Server.

Figure 2–1 Active Directory Connector Deployment Architecture with the .NET Connector Server



- **Machine 1** has Oracle Waveset deployed.
- **Machine 2** has the Active Directory connector bundle deployed in the .NET Connector Server. The .NET Connector Server is part of the Identity Connector Framework (ICF). The machine where the .NET Connector Server is installed must also have Microsoft .NET Framework 3.5 or later installed.
- **Machine 3** has the Windows target system deployed with either Active Directory or Active Directory Lightweight Directory Services (AD LDS).

> **Note:** In this scenario, Machine 2 and Machine 3 must be in the same domain.
>
> If you prefer, you can also install the .NET Connector Server and the Active Directory connector bundle on the Windows target system. This machine must also have Microsoft .NET Framework 3.5 or later installed.

Oracle Waveset communicates directly with the Identity Connector Framework (ICF), which passes requests to the .NET Connector Server over the network.

The .NET Connector Server then serves as a proxy to provide access to the Active Directory connector, which is deployed within the .NET Connector Server. The results of operations performed on the connector are passed back to Oracle Waveset via the ICF.

2.3.2 Preinstallation Tasks for the Active Directory Connector

Before you install the Active Directory connector, install and configure the .NET Connector Server, and configure the target system as follows:

- Downloading the Active Directory Connector
- Considering the .NET Connector Server Installation Locations
- Installing, Configuring, and Enabling Logging on the .NET Connector Server
- Delegating Control of Organizational Units and Custom Object Classes

2.3.2.1 Downloading the Active Directory Connector

The Active Directory connector is available on the Oracle Identity Manager Connector Downloads page:

`http://www.oracle.com/technetwork/middleware/id-mgmt/downloads/connectors-101674.html`

2.3.2.2 Considering the .NET Connector Server Installation Locations

> **Note:** This section applies only when the Active Directory connector is configured to contact an Active Directory server and **not** for AD LDS.

Unless the Active Directory Domain Controller Hostname (`LDAPHostName`) resource attribute is set, the connector will perform a serverless bind to the directory. For the serverless bind to work, the .NET Connector Server must be installed on a system that is in a domain and that knows about the domain and directory to be managed. All Windows domains managed by a connector must be part of the same forest. Managing domains across forest boundaries is unsupported. If you have multiple forests, install at least one .NET Connector Server in each forest.

The Active Directory Domain Controller Hostname resource attribute tells the connector to bind to a particular DNS hostname or IP address. This is the opposite of a serverless bind. However, the Active Directory Domain Controller Hostname does not necessarily have to specify a specific domain controller. The DNS name of an Active Directory domain can be used. If the connector's DNS server is configured to return multiple IP addresses for that DNS name, then one of them will be used for the directory bind. This avoids having to rely on a single domain controller.

All operations require that the .NET Connector Server be a member of a domain.

2.3.2.3 Installing, Configuring, and Enabling Logging on the .NET Connector Server

Although the .NET Connector Server can be executed as a standalone Windows application, it is recommended that you install the .NET Connector Server using the installation package (`ServiceInstall.msi`). This package registers the .NET Connector Server as a Windows service and then automatically starts the service.

The machine where the .NET Connector Server is installed must also have Microsoft .NET Framework 3.5 or later installed.

This section discusses the following topics:

- Installing and Configuring the .NET Connector Server

- Enabling Logging for the Active Directory Connector
- Configuring Log File Rotation

2.3.2.3.1 Installing and Configuring the .NET Connector Server

To install and configure the .NET Connector Server, follow these steps:

1. To install the .NET Connector Server, execute `ServiceInstall.msi` and follow the wizard. The wizard takes you through the installation process step-by-step. After completion, the .NET Connector Server is registered as a Windows service.

   The `ServiceInstall.msi` file is included as part of the Oracle Waveset patch shown in Certified Components for the Active Directory Connector.

2. Start the Microsoft Services Console.
3. If the .NET Connector Server is running, stop it by stopping the Windows service.
4. To set a custom key for the .NET Connector Server, use the `/setkey` command-line argument, as follows:

   a. Change to the directory where the .NET Connector Server was installed. The default directory is:

      `C:\Program Files\Identity Connectors\Connector Server`

   b. Execute the following command:

      `ConnectorServer.exe /setkey` *`newkey`*

      where *`newkey`* is the value for the new key.

      This key is required by any client that connects to this .NET Connector Server.

5. Check the settings in the .NET Connector Server configuration file (`ConnectorServer.exe.config`). These settings are in the tag named `AppSettings`. For example:

```
<add key="connectorserver.port" value="8759" />
<add key="connectorserver.usessl" value="false" />
<add key="connectorserver.certificatestorename"
    value="ConnectorServerSSLCertificate" />
<add key="connectorserver.ifaddress" value="0.0.0.0" />
```

   The most common settings you might want to change are:

   - Port number: To change the port, set `connectorserver.port` to a value other than 8759.
   - SSL settings: To use SSL, set `connectorserver.usessl` to `true` and then set `connectorserver.certificatestorename` to your certificate store name.
   - Listening socket bind: To change the listening socket bind, set `connectorserver.ifaddress` to an address other than `0.0.0.0`.
   - Trace settings: To set trace settings, see Enabling Logging for the Active Directory Connector.

6. Save the following configuration information from the .NET Connector Server installation for later use:

   - Host name or IP address
   - Connector Server port

- Connector Server key values
- Whether SSL is enabled

7. When you are finished configuring the .NET Connector Server, restart it by restarting the Windows service. Or, you can also restart the .NET Connector Server using the following command:

```
ConnectorServer.exe /run
```

2.3.2.3.2 Enabling Logging for the Active Directory Connector

The Active Directory connector uses the built-in logging mechanism of the .NET framework. Logging for the Active Directory connector is not integrated with Oracle Waveset. The logging level is set in the .NET Connector Server configuration file (`ConnectorServer.exe.config`).

To enable logging for the Active Directory connector, follow these steps:

1. Go to the directory where the `ConnectorServer.exe` file is installed. The default directory is `C:\Program Files\Identity Connectors\Connector Server`.

   The `ConnectorServer.exe.config` file should be present in this directory.

2. In the `ConnectorServer.exe.config` file, add the following snippet, shown in bold text:

```
<system.diagnostics>
  <trace autoflush="true" indentsize="4">
    <listeners>
      <remove name="Default" />
      <add name="myListener" type="System.Diagnostics.TextWriterTraceListener"
           initializeData="c:\connectorserver2.log"
traceOutputOptions="DateTime">
        <filter type="System.Diagnostics.EventTypeFilter"
initializeData="Information" />
      </add>
    </listeners>
  </trace>
  <switches>
    <add name="ActiveDirectorySwitch" value="4" />
  </switches>
</system.diagnostics>
```

   The `value="4"` sets the logging level to Verbose. This value can be set as follows:

Value	Logging Level
`value="4"` or `value="Verbose"`	Verbose level. Most granular.
`value="3"` or `value="Information"`	Information level.
`value="2"` or `value="Warning"`	Warning level.
`value="1"` or `value="Error"`	Error level.
`value="0"`	No logging.

   However, remember that the logging level has a direct effect on the performance of the .NET Connector Server.

3. After you make the configuration change, stop and then restart the .NET Connector Server service. Or, you can also restart the .NET Connector Server using the following command:

```
ConnectorServer.exe /run
```

2.3.2.3.3 Configuring Log File Rotation Information about events that occur during the course of reconciliation and provisioning operations are stored in a log file. As you use the connector over a period time, the amount of information written to a log file increases. If no rotation is performed, then log files become huge.

To avoid such a scenario, perform the procedure described in this section to configure rotation of the log file.

To configure rotation of a log file on a daily basis:

1. Log in to the computer that is hosting the connector server.
2. Stop the Connector Server.
3. Back up the ConnectorServer.exe.config file. The default location of this file is `C:\Program Files\Identity Connectors\Connector Server.`
4. In a text editor, open the `ConnectorServer.exe.config` file for editing.
5. Search for the `<listeners>` and `</listeners>` elements and replace the text between these elements with the following:

```
<remove name="Default" />
<add name="FileLog"
type="Microsoft.VisualBasic.Logging.FileLogTraceListener,Microsoft.VisualBasic,
Version=8.0.0.0,Culture=neutral,PublicKeyToken=b03f5f7f11d50a3a"
initializeData="FileLogWriter"
traceOutputOptions="DateTime"
BaseFileName="ConnectorServerDaily"
Location="Custom"
CustomLocation="C:\ConnectorServerLog\"
LogFileCreationSchedule="Daily">
<filter type="System.Diagnostics.EventTypeFilter"
initializeData="Information"/>
</add>
```

6. Save and close the file.
7. Start the Connector Server.

See Also: The following URL for more information about configuring log file rotation:

http://msdn.microsoft.com/en-us/library/microsoft.visualbasic.logging.filelogtracelistener.aspx

2.3.2.4 Delegating Control of Organizational Units and Custom Object Classes

By default, user accounts that belong to the Account Operators group can manage only user and group objects. To manage organizational units or custom object classes, you must assign the necessary permissions to a user account. In other words, you must delegate complete control for an organizational unit or custom object class to a user or group object. In addition, you need these permissions to successfully perform provisioning of custom object classes.

This is achieved by using the Delegation of Control Wizard. An example for managing organizational units is creating organizational units.

To delegate control for an organizational unit or custom object class to a user account:

Note: In a parent-child deployment environment or forest topology, perform this procedure on all the child domains.

1. In the Active Directory Users and Computers window, in the navigation tree, right-click the organizational unit whose control you want to delegate, and then click **Delegate Control.**

   The Delegation of Control Wizard is displayed.

   Note: If you want to delegate control for all organization units under the root context, then delegate control at the root context level.

2. On the Welcome to the Delegation of Control Wizard page, click **Next.**
3. On the Users or Groups page, to select either a user or group to whom you want to delegate control:
   a. **Click Add.**
   b. In the Select Users, Computers, or Groups dialog box, enter a user or group name. For example, enter `OIMUser`.
   c. Click **Check Names.**
   d. Click **OK** to close the dialog box.
4. Click **Next.**
5. On the Tasks to Delegate page, select the **Create a custom task to delegate** option, and then click **Next.**
6. On the Active Directory Object Type page, select **Only the following objects in the folder**, and then select **Organization Unit Objects.** If you are delegating control for custom object classes, then select the custom object class for which you want to delegate control.
7. Select the **Create selected objects in the folder** and **Delete selected objects in the folder** options, and then click **Next.**
8. On the Permissions page:
   - For Organizational Units, select **Full Control,** click **Next,** and then click **Finish.**
   - For custom object classes, select the required permissions, click **Next,** and then click **Finish.**

2.3.3 Installing the Active Directory Connector

It is recommended that you install the Active Directory connector bundle in the .NET Connector Server, as follows:

1. Change to the directory where the .NET Connector Server was installed.
2. Unzip the Active Directory connector ZIP file in the directory from Step 1.

3. Restart the .NET Connector Server service. Or, you can also restart the .NET Connector Server using the following command:

```
ConnectorServer.exe /run
```

2.3.4 Postinstallation Tasks for the Active Directory Connector

- Creating an Active Directory Connector Resource
- Adding Attributes to Active Directory Connector Resource Forms
- Adding Byte[] Datatype Attribute to Active Directory Connector Resource Forms
- Displaying Group Names for Active Directory LDS or ADAM Resources
- Configuring Before and After Actions for the Active Directory Connector
- Passing Process Form Parameters to Scripts

2.3.4.1 Creating an Active Directory Connector Resource

To create an Active Directory connector resource, follow these steps:

1. Log in to the Oracle Waveset Administrator interface.
2. Create the Active Directory connector resource by following the Create Windows Active Directory Connector Resource wizard.
3. Select the Active Directory Connector Version as "1.1.0.6380".
4. Select the .NET Connector Server on which the Active Directory connector bundle is deployed.
5. Specify values for the Active Directory connector, depending on your deployment. For more information, see:
   - Active Directory Connector Resource Configuration Parameters
   - Object Classes and Attributes Supported by the Active Directory Connector
6. To enable checking the password history for an Active Directory account when end-users change their password, the WS_USER_PASSWORD attribute is available in the Active Directory connector userForm.xml file. This attribute is ignored if the PasswordNever Expires field is set for the resource.

2.3.4.2 Adding Attributes to Active Directory Connector Resource Forms

The Active Directory connector Group, Organizational unit, and Container resource forms are customizable. You can add attributes to these forms, as required by your deployment.

For example, when you provision an organizational unit, the Name, Short Name, and Description attributes are available for the organization. However, when you create a resource object of type Organization, by default you see only the Name attribute.

Therefore, to add additional attributes to provision an organizational unit, edit the Windows Active Directory Create Organizational Unit Form, as follows:

1. Go to the Oracle Waveset debug page:

   `http://host_name:port/idm/debug`

2. Select Resource Form from the drop-down box, which is adjacent to List Objects, and then click **List Objects.**

3. Select the Windows Active Directory Create Organizational Unit Form from the list and click **Edit.**
4. Add additional attributes to the form. For example:

```
<Field name='organizational unit.attributes.description'>
  <Display class='Text'>
    <Property name='title' value='Description:'/>
  </Display>
</Field>
```

Add any other target attributes to the Windows Active Directory Create Organizational Unit Form by following these same steps.

You can also add attributes to the Windows Active Directory Create Group Form and Windows Active Directory Create Container Form, as required by your deployment.

2.3.4.3 Adding Byte[] Datatype Attribute to Active Directory Connector Resource Forms

You can add attributes of Byte[] datatype to connector resource forms such as User and Group, depending on your requirement. For example, you can add the `thumbnailPhoto` attribute to the User resource form as follows:

1. In a text editor, open the `owglue\sample\connectors\ActiveDirectoryConnector-idmglue\UserForm.xml` file located in the installation media.
2. Add additional attributes to the form. For example:

```
<Field name='accounts[$(RESOURCE_NAME)].thumbnailPhoto'>
   <Expansion>
      <cond>
         <isnull>
            <ref>FileSource</ref>
         </isnull>
         <ref>accounts[$(RESOURCE_NAME)].thumbnailPhoto</ref>
         <new class='com.waveset.util.Binary'>
            <ref>FileSource</ref>
         </new>
      </cond>
   </Expansion>
</Field>
<Field name='FileSource'>
  <Display class='FileUpload'>
    <Property name='title' value='Thumbnail Photo'/>
  </Display>
</Field>
```

3. Import the User form as follows:
   a. Click the Configure tab.
   b. Click **Import Exchange File.**
   c. Select the User resource form and then click **Import.**
4. Add the `thumbnailPhoto` attribute as follows:
   a. Click the Resource tab.
   b. Click **Edit Resource Page,** to open the resource to which you want to add the attribute.

The Edit *RESOURCE_NAME* Resource Wizard page is displayed.

c. On the Resource Parameters page, click **Next.**

d. On the Account Attributes page, click **Add Attribute.**

e. In the new row that is added to the table, specify the following values:

   – In the Identity System User Attributes column, enter `thumbnailPhoto` in the text field.

   – In the Attribute Type column, from the drop-down list, select **binary.**

   – In the Resource User Attribute column, enter `thumbnailPhoto`.

f. Click **Save.**

2.3.4.4 Displaying Group Names for Active Directory LDS or ADAM Resources

If you create an Active Directory LDS (AD LDS) or ADAM resource, groups for the resource are shown in Oracle Waveset without the group name for both provisioning and reconciliation.

To get the group name to display for an AD LDS or ADAMresource, edit the resource forms, as follows:

- In the Windows Active Directory Create Group Form, replace all occurrences of samAccountName with cn.
- In all other Active Directory connector resource forms, including the Windows Active Directory Update Group Form, replace samAccountName with userPrincipalName. Any references to samAccountName in these forms can cause undesirable behavior for an AD LDS or ADAMresource.

2.3.4.5 Configuring Before and After Actions for the Active Directory Connector

This section describes how to configure a "create after action" for the Active Directory connector, but the steps apply to all types of actions as well.

To configure a "create after action" for the Active Directory connector, follow these steps:

1. Import a resource action similar to the following example:

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE Waveset PUBLIC 'waveset.dtd' 'waveset.dtd'>
<Waveset>
<ResourceAction name='ADAfterCreate'>
<ResTypeAction restype='Windows Active Directory' timeout='6000'
    actionType='SHELL' execMode='resource'>
<act>
    echo create>> C:\Temp\%WSUSER_accountId%.txt
    exit
</act>
</ResTypeAction>
</ResourceAction>
</Waveset>
```

   Note. `execMode` can be set to either `resource` or `connector`; however, only `resource` is supported by the Active Directory connector. If `execMode` is set to `connector`, it will be mapped to the generic `ScriptOnConnector` SPI operation provided by the Connector Framework.

2. In Oracle Waveset, add the `create after action` attribute to the Identity System User Attribute column on the Active Directory resource.
3. Type `IGNORE_ATTR` in the Resource User Attribute column and leave the other values as the defaults. Then save the changes.
4. Add a form field to the resource specific user form (probably the AD IdC User Form) with the name as `create after action`. For example:

```
<Field name='accounts[$(RESOURCE_NAME)].create after action'>
<Expansion>
<s>ADAfterCreate</s>
</Expansion>
</Field>
```

5. You can now test the script.

2.3.4.6 Passing Process Form Parameters to Scripts

This section describes how you can pass Active Directory connector process form parameters to scripts during the execution of before and after scripts. Visual Basic (VB), batch, and Perl scripts function similarly and can execute any commands that can be executed on the target system command line or shell.

The following example shows a Visual Basic script that consumes data dynamically from the process form. This is an example procedure for an After Create action, which requires creating a user also in an organizational unit other than the organizational unit where the user is provisioned.

1. Create a script file on the Oracle Waveset machine. For example:

```
C:\arg.vbs %givenName%
```

   Note: There is a space between C:\arg.vbs and %givenName%.

2. On the machine hosting the target system, create the arg.vbs file in the C:\ directory.
3. Include the following lines in the arg.vbs file:

```
Set args = WScript.Arguments
GivenNameFromArg = args.Item(0)
lengthGivenName = Len(GivenNameFromArg) - 2
GivenNameTrim = Mid(GivenNameFromArg, 2, lengthGivenName)
Set objOU =
GetObject("LDAP://ldapserver.example.com:389/OU=org,dc=example,dc=com")
Set objUser = objOU.Create("User", "cn=scriptCreate" & GivenNameTrim )
objUser.Put "givenName", "scriptCreate" & GivenNameTrim
objUser.Put "sAMAccountName", "scriptCreate " & GivenNameTrim
objUser.Put "userPrincipalName", "scriptCreate" & GivenNameTrim
objUser.Put "displayName", "scriptCreate" & GivenNameTrim
objUser.Put "sn", "scriptCreate" & GivenNameTrim
objUser.SetInfo
```

4. Save and close the arg.vbs file.
5. Provision a user account on Oracle Waveset.

2.3.5 Configuring the Active Directory Connector for AD LDS

The Active Directory connector supportsMicrosoft Active Directory Lightweight Directory Services (AD LDS), formerly called Active Directory ApplicationMode (ADAM). To configure the Active Directory connector for AD LDS, follow these steps:

1. Go to the Oracle Waveset debug page:

   ```
   http://host_name:port/idm/debug
   ```

2. Select Resource from the drop-down box adjacent to List Objects, and then click on List Objects.

3. Edit the AD LDS resource, as follows:

   a. Modify the mapping of accountID to UserPrincipalName. The accountID is mapped to sAMAccountName. Because sAMAccountName is not present in AD LDS, update the following line by specifying `mapName='UserPrincipalName'`:

   ```
   <AccountAttributeType id='15' name='accountId'
   syntax='string' mapName='sAMAccountName' mapType='string'>
   ```

   b. Modify the Group Object type by specifying cn as displayNameAttr instead of sAMAccountName. In the following lines, by replace samAccountName with cn:

   ```
   <ObjectAttributes idAttr='distinguishedName'
   displayNameAttr='samAccountName' descriptionAttr='description'
   objectClassAttr='objectclass'>
   <ObjectAttribute name='samAccountName' type='string'/>
   ```

4. Edit the Resource Form (Windows Active Directory Create Group Form) by selecting the Resource Form from the drop-down adjacent to List Objects. Then, replace all references to sAMAccountName with cn.

2.3.6 Enabling Reconciliation and Provisioning Operations Across Multiple Domains

The Active Directory connector supports reconciliation and provisioning operations across multiple domains in a single forest. Reconciliation runs are performed by using the Global Catalog Server and provisioning operations are performed by using LDAP referrals. If you want to enable reconciliation and provisioning across multiple domains, then perform the procedure described in the following sections:

- Enabling Reconciliation Across Multiple Domains
- Enabling Provisioning Across Multiple Domains

2.3.6.1 Enabling Reconciliation Across Multiple Domains

To perform reconciliation across multiple domains, this connector uses both the domain controller and the Global Catalog Server for fetching records from the target system.

During reconciliation, records from the Global Catalog Server are fetched to the connector. After a record is fetched into the connector, the distinguishedName and uSNChanged attribute values are read. By using the distinguishedName, the connector performs an LDAP query on the domain controller that contains the actual data (referrals are used here). This approach is used for reconciliation because the Global Catalog Server has only partial set of records. Complete data can only be fetched from the domain controller.

After all records are fetched, Oracle Waveset keeps track of the maximum value of the uSNChanged attribute of a domain controller on which the Global Catalog Server is running. In incremental mode, only records whose uSNChanged attribute values are greater than current value in the Latest Token attribute are fetched from the Global Catalog Server.

Therefore, any updates made to a record on the target system must update the uSNChanged attribute of that record in the Global Catalog Server so that the connector can detect records that have been updated since the last reconciliation run and then fetch them into Oracle Waveset.

To enable reconciliation across domains, follow these steps:

1. Set the value of the Search Child Domains entry to yes.
2. Specify the name of the domain controller that is hosting the Global Catalog Server as the value of the SyncGlobalCatalogServer in the resource configuration.

Note: While performing group reconciliation in a cross-domain environment, the connector fetches only those groups of the account that are visible to the domain controller on which the account is present.

2.3.6.2 Enabling Provisioning Across Multiple Domains

In a parent-child deployment environment of the target system, before performing provisioning operations across multiple domains, it is expected that the target system resource is configured with the parent domain. In a replication environment of the target system, before performing provisioning operations across multiple domains, it is expected that the target system resource is configured with any of the domain controllers.

This scenario is illustrated by the following example.

Suppose a parent-child domain environment in which the parent domain is dc1 and child domain is dc2. The target system resource is configured to include dc1 as the value of the LDAPHostName parameter the name of the parent domain as the value of theDomainName parameter.

During provisioning, if you select an organization that belongs to the child domain, multiple groups that span across domains, and the manager from the parent domain, then LDAP referrals are internally used by ADSI (Active Directory Service Interfaces). This is because all connectors operations are leveraged to ADSI, which enables creation of an account in the child domain even without providing any details of the child domain in the Resource Configuration.

All this information is internally calculated depending upon the organization that is selected during the provisioning operation. In the connector, the referral chasing option is set to All, which means that all referrals are chased when any referral is provided by the domain controller. Therefore, no explicit configuration procedure is required to enable provisioning across multiple domains.

For more information, see the ADSI documentation about LDAP referrals.

2.3.7 Adding Auxiliary Classes to Users

To perform the procedure described in this section, all domain controllers in the forest must be running Microsoft Windows Server 2003 or later, and the forest functional mode must be Microsoft Windows Server 2003 or later. For more information on dynamic auxiliary object classes, see "Dynamically Linked Auxiliary Classes (Windows)" at the following Web site:

`http://msdn.microsoft.com/en-us/library/windows/desktop/ms676289%28v=vs.85%29.aspx`

The following is the procedure to add auxiliary classes to users:

1. Create an entry for the `AccountObjectClass` attribute in the `owglue\sample\connectors\ActiveDirectoryConnector-idmglue\ResourceWizard.xml` file that is located in the connector installation media.
2. If the auxiliary class has mandatory attributes, then create an entry for the `ObjectClassMandatoryAttributes` attribute and the mandatory attributes in the `owglue\sample\connectors\ActiveDirectoryConnector-idmglue\ResourceWizard.xml` file that is located in the connector installation media. Ensure to set the value of the Display class element to `MultiSelect` when you create this entry.
3. Save the file.
4. Import the `ResourceWizard.xml` file as follows:
   a. Click the Configure tab.
   b. Click **Import Exchange File.**
   c. Select the **ResourceWizard.xml** file and then click **Import.**

To add the auxiliary class to the resource in Oracle Waveset:

Note: To explain this procedure, it has been assumed that CustomAuxClass is an auxiliary class with the following attributes:

- CustomAttribute1

  This is a mandatory attribute.
- CustomAttribute2

  This is an optional attribute.
- CustomAttribute3

  This is a mandatory attribute.

---

1. Open an Active Directory resource.
2. Enter the name of the auxiliary class in the column corresponding to the `AccountObjectClass` attribute.

   For example, enter `CustomAuxClass`.
3. From the list of attributes displayed in the colum corresponding to the `ObjectClassMandatoryAttributes` attribute, select the mandatory attributes of the auxiliary class, and then move it to the right column.

   For example, select `CustomAttribute1` and `CustomAttribute3` and in the left column and move it to the right column.
4. Click **Next.**
5. In the Account Attributes tab, add all the attributes of the auxiliary class. For example, you must add the `CustomAttribute1, CustomAttribute2,` and `CustomAttribute3` attributes.
6. Click **Save.**

To display the custom attributes on the user form in Oracle Waveset:

1. Go to the Oracle Waveset debug page:

`http://host_name:port/idm/debug`

2. In the column corresponding to List Objects, select **User Form.**
3. Click **List Objects.**
4. Click the edit button corresponding to the User Form (for example, AD IdC User Form).
5. Add the following lines for each custom attribute:

```
<Field name='accounts[$(RESOURCE_NAME)].<AttributeName>'>
          <Display class='Text'>
            <Property name='title' value="<AttributeName>"/>
            <Property name='size' value='25'/>
          </Display>
        </Field>
```

The following is a sample of code that you must add for the `CustomAuxClass` auxiliary class:

```
<Field name='accounts[$(RESOURCE_NAME)].CustomAttribute1'>
          <Display class='Text'>
            <Property name='title' value=" CustomAttribute1"/>
            <Property name='size' value='25'/>
          </Display>
        </Field>
        <Field name='accounts[$(RESOURCE_NAME)].CustomAttribute2'>
          <Display class='Text'>
            <Property name='title' value=" CustomAttribute2"/>
            <Property name='size' value='25'/>
          </Display>
        </Field>
```

6. Click **Save.**

2.3.8 Adding Custom Object Classes

This connector supports adding custom object classes to users. The custom object class has the attributes of the user and custom attributes.

The following is the procedure to include a custom object class:

Note: To explain this procedure, it has been assumed that `CustomObjectClass` is a custom object class with the following attributes:

- `CustomStringAttr, CustomIntAttr`

  These are mandatory attributes.

- `SecondCustomStringAttr`

  This is an optional attribute.

---

1. If the custom object class has mandatory attributes, then create an entry for the `ObjectClassMandatoryAttributes` attribute in the `owglue\sample\connectors\ActiveDirectoryConnector-idmglue\ResourceWizard.xml` file that is located in the connector installation media. Ensure to set the value of the Display class element to `MultiSelect` when you create this entry.

The following is a sample of code to add the `ObjectClassMandatoryAttributes` attribute and set the Display class element to `Multiselect`:

```
<Field name="resourceAttributes[ObjectClassMandatoryAttributes].value"
required="false">
    <Display class="MultiSelect">
        <Property name="title" value="ObjectClassMandatoryAttributes"/>
        <Property name="allowedValues">
            <List>
                <String>CustomStringAttr</String>
                <String>CustomIntAttr</String>
            </List>
        </Property>
    </Display>
</Field>
```

2. Save the file.
3. Import the `ResourceWizard.xml` file as follows:
   a. Click the Configure tab.
   b. Click **Import Exchange File.**
   c. Select the **ResourceWizard.xml** file and then click **Import.**

To add the custom object class to the resource in Oracle Waveset:

Note: To explain this procedure, it has been assumed that CustomAuxClass is an auxiliary class with the following attributes:

- `CustomAttribute1`

  This is a mandatory attribute.

- `CustomAttribute2`

  This is an optional attribute.

- `CustomAttribute3`

  This is a mandatory attribute.

---

1. Open an Active Directory resource.
2. Enter the name of the custom object class in the column corresponding to the `Object Class for User Objects` attribute.

   For example, enter `CustomObjectClass`.
3. Click **Next.**
4. In the Account Attributes tab, add all the attributes of the custom object class. For example, you must add the `CustomStringAttr, CustomIntAttr` and `SecondCustomStringAttr` attributes.
5. Click **Save.**

To display the custom attributes on the user form in Oracle Waveset:

1. Go to the Oracle Waveset debug page:

   `http://`*`host_name`*`:`*`port`*`/idm/debug`
2. In the column corresponding to List Objects, select **User Form.**

3. Click **List Objects.**
4. Click the edit button corresponding to the user form (for example, AD IdC User Form).
5. Add the following lines for each custom attribute:

```
<Field name='accounts[$(RESOURCE_NAME)].<AttributeName>'>
          <Display class='Text'>
            <Property name='title' value="<AttributeName>"/>
            <Property name='size' value='25'/>
          </Display>
        </Field>
```

The following is a sample of code that you must add for the `CustomObjectClass` auxiliary class:

```
<Field name='accounts[$(RESOURCE_NAME)].CustomStringAttr'>
          <Display class='Text'>
            <Property name='title' value="CustomStringAttr"/>
            <Property name='size' value='25'/>
          </Display>
        </Field>
<Field name='accounts[$(RESOURCE_NAME)].SecondCustomStringAttr'>
          <Display class='Text'>
            <Property name='title' value="SecondCustomStringAttr"/>
            <Property name='size' value='25'/>
          </Display>
        </Field>
<Field name='accounts[$(RESOURCE_NAME)].CustomIntAttr'>
          <Display class='Text'>
            <Property name='title' value="CustomIntAttr"/>
            <Property name='size' value='25'/>
          </Display>
        </Field>
```

6. Click **Save.**

2.4 Using the Active Directory Connector

2.4.1 Active Directory Usage Considerations

This section lists dependencies and limitations related to using the Active Directory connector, including the following section:

2.4.1.1 Specifying a Domain for Pass-Through Authentication

Note: This section applies only when the Active Directory connector is configured to contact an Active Directory server and **not** for AD LDS.

In a default configuration, pass-through authentication is accomplished by sending the user ID and password only. These two attributes are configured in the `AuthnProperties` element in the resource object's XML as `w2k_user` and `w2k_password`. Without a domain specification, the Active Directory connector searches all known domains and tries to authenticate the user in the domain that contains the user.

In a trusted multi-domain environment, there can be two possible situations:

- All domains contain a synchronized user and password combination.
- The user/password combination is domain dependent.

When the user/password combination is synchronized, configure your Active Directory resources so that they are common resources.

For more information about setting up a common resource, see the *Oracle Waveset 8.1.1 Business Administrator's Guide* in the following library:

`http://docs.oracle.com/cd/E19225-01/index.html`

In an environment with multiple trusted domains and Active Directory forests, the authentication can fail using any of these configurations because the Global Catalog does not contain cross-forest information. If a user supplies a wrong password, it could also lead to account lockout in the user's domain if the number of false attempts is greater than the lockout threshold.

Login failures will occur in domains if the user exists in the domain and the password is not synchronized.

It is not possible to use multiple data sources for the domain information in one Login Module Group.

2.4.2 Object Classes and Attributes Supported by the Active Directory Connector

This section provides the following information about the object classes and attributes supported by the Active Directory connector:

- `__ACCOUNT__` Object Class for the Active Directory Connector
- `__GROUP__` (Group) Object Class for the Active Directory Connector
- `organizationalUnit` Object Class for the Active Directory Connector
- Attribute Syntax Support for the Active Directory Connector

> **Note:** If you wish, you can change the provisioning or reconciliation attribute map by adding arbitrary attributes (using the supported attribute types) defined in the Active Directory schema on the object class. You can also remove non-operational attributes.
>
> The Active Directory connector also supports custom object classes and different object classes for different types of users. The connector supports the provisioning and reconciliation for custom object classes. For example, you might create a custom object class such as `ObjectClass1`, extending the USER.

2.4.2.1 `__ACCOUNT__` Object Class for the Active Directory Connector

Unless noted otherwise, an attribute is single-valued and optional, and can be created, updated, and read.

Note: When you perform group reconciliation for the first time, the connector fetches all groups from the target system and stores it in Oracle Waveset cache. From this point onward, whenever you open the user form, all groups are loaded from the cache. Whenever you add a new group in the target system, then you must clear cache for the new group to reflect in Oracle Waveset. Similarly, everytime you create a new resource, you must clear cache. To clear the cache, navigate to debug page and click **Clear Resource Object List Cache.**

Table 2–5 *__ACCOUNT__ Object Class Attributes for the Active Directory Connector*

Attribute Name	Type	Description
sAMAccountName	String	For AD DS only; not for AD LDS.
givenName	String	-
sn	String	-
displayName	String	-
mail	String	-
telephoneNumber	String	-
employeeID	String	-
division	String	-
mobile	String	-
middleName	String	-
description	String	Multi-valued.
department	String	-
manager	String	-
title	String	-
initials	String	-
co	String	-
company	String	-
facsimileTelephoneNumber	String	-
homePhone	String	-
streetAddress	String	-
l	String	-
st	String	-
postalCode	String	-
TerminalServicesInitialProgram	String	For AD DS only; not for AD LDS.
TerminalServicesWorkDirectory	String	For AD DS only; not for AD LDS.
AllowLogon	Integer	For AD DS only; not for AD LDS.
MaxConnectionTime	Integer	For AD DS only; not for AD LDS.
MaxDisconnectionTime	Integer	Cannot be created or updated. For AD DS only; not for AD LDS.

Table 2–5 (Cont.) __ACCOUNT__ Object Class Attributes for the Active Directory

Attribute Name	Type	Description
MaxIdleTime	Integer	For AD DS only; not for AD LDS.
ConnectClientDrivesAtLogon	Integer	Cannot be created or updated. For AD DS only; not for AD LDS.
ConnectClientPrintersAtLogon	Integer	Cannot be created or updated. For AD DS only; not for AD LDS.
DefaultToManPrinter	Integer	Cannot be created or updated. For AD DS only; not for AD LDS.
BrokenConnectionAction	Integer	Cannot be created or updated. For AD DS only; not for AD LDS.
ReconnectionAction	Integer	Cannot be created or updated. For AD DS only; not for AD LDS.
EnableRemoteControl	Integer	Cannot be created or updated. For AD DS only; not for AD LDS.
TerminalServicesProfilePath	String	Cannot be created or updated. For AD DS only; not for AD LDS.
TerminalServicesHomeDirectory	String	Cannot be created or updated. For AD DS only; not for AD LDS.
TerminalServicesHomeDrive	String	Cannot be created or updated. For AD DS only; not for AD LDS.
uSNChanged	String	Cannot be created or updated.
ad_container	String	Cannot be created or updated.
otherHomePhone	String	Multi-valued.
distinguishedName	String	Cannot be created or updated.
objectClass	String	Cannot be created or updated.
homeDirectory	String	For AD DS only; not for AD LDS.
PasswordNeverExpires	Boolean	-
dynamicAuxClasses	String	Multi-valued. Not readable and not returned by default. Can be created only.
__ENABLE__	Boolean	-
__LOCK_OUT__	Boolean	-
__PASSWORD_EXPIRED__	Boolean	-
__CURRENT_PASSWORD__	GuardedString	-
__PASSWORD__	GuardedString	Multi-valued. Not readable and not returned by default.
__GROUPS__	String	Multi-valued.
__DESCRIPTION__	String	-
__SHORT_NAME__	String	-
__NAME__	String	Required.
PasswordNotRequired	Boolean	Cannot be read.
whenChanged	Long	Cannot be created or updated.

Table 2–5 (Cont.) __ACCOUNT__ Object Class Attributes for the Active Directory

Attribute Name	Type	Description
__UPN_WO_DOMAIN__	String	Cannot be created or updated and not returned by default.
__PARENTCN__	String	Cannot be created or updated and not returned by default.

2.4.2.2 ___GROUP___ (Group) Object Class for the Active Directory Connector

The Active Directory connector supports the attributes shown in the following table by default. Support for other attributes is also provided by the Active Directory connector. To include additional attributes, add the desired attributes to the ADgroupcreate.xml form and then import the revised form into Oracle Waveset. For more information see "Adding Attributes to Active Directory Connector Resource Forms".

Unless noted otherwise, an attribute is single-valued and optional, and can be created, updated, and read.

Table 2–6 __GROUP__ (Group) Object Class Attributes for the Active Directory Connector

Attribute Name	Type	Description
samAccountName	String	For AD DS only; not for AD LDS.
description	String	-
managedby	String	-
mail	String	For AD DS only; not for AD LDS.
groupType	Integer	-
member	String	Multi-valued. Not readable and not returned by default.

2.4.2.3 organizationalUnit Object Class for the Active Directory Connector

The Active Directory connector supports the attributes shown in the following table by default. Support for other attributes is also provided by the Active Directory connector. To include additional attributes, add the desired attributes to the ADorganizationalunitcreate.xml form and then import the revised form into Oracle Waveset. For more information see Adding Attributes to Active Directory Connector Resource Forms.

> **Note:** For the Active Directory connector to provision an organizational unit, an organization must already exist in the Active Directory or AD LDS target resource. Otherwise, the Active Directory connector supports the provisioning of sub-organizational units only.

Unless noted otherwise, an attribute is single-valued and optional, and can be created, updated, and read.

Table 2–7 `organizationalUnit` Object Class Attributes for the Active Directory Connector

Attribute Name	Type	Description
`ou`	String	Name of the organizational unit. Cannot be created or updated.
`__DESCRIPTION__`	String	Description of the organizational unit.

2.4.2.4 Attribute Syntax Support for the Active Directory Connector

The syntax (or type) of an attribute usually determines whether an attribute is supported. In general, Oracle Waveset supports Boolean, string, and integer syntaxes. Binary strings and similar syntaxes are not supported. This section provides the following information:

- Active Directory Syntaxes Supported by Oracle Waveset
- Active Directory Syntaxes Not Supported by Oracle Waveset

2.4.2.4.1 Active Directory Syntaxes Supported by Oracle Waveset The following table lists the Active Directory syntaxes supported by Oracle Waveset:

Table 2–8 Active Directory Syntaxes Supported by Oracle Waveset

Active Directory Syntax	Waveset Syntax	Syntax ID	OM ID	ADS Type
`Boolean`	Boolean	2.5.5.8	1	`ADSTYPE_BOOLEAN`
`Enumeration`	String	2.5.5.9	10	`ADSTYPE_INTEGER`
`Integer`	Integer	2.5.5.9	2	`ADSTYPE_INTEGER`
`DN String`	String	2.5.5.1	127	`ADSTYPE_DN_STRING`
`Presentation Address`	String	2.5.5.13	127	`ADSTYPE_CASE_IGNORE_STRING`
`IA5 String`	String	2.5.5.5	22	`ADSTYPE_PRINTABLE_STRING`
`Printable String`	String	2.5.5.5	19	`ADSTYPE_PRINTABLE_STRING`
`Numeric String`	String	2.5.5.6	18	`ADSTYPE_NUMERIC_STRING`
`OID String`	String	2.5.5.2	6	`ADSTYPE_CASE_IGNORE_STRING`
`Case Ignore String` (teletex)	String	2.5.5.4	20	`ADSTYPE_CASE_IGNORE_STRING`
`Unicode String`	String	2.5.5.12	64	`ADSTYPE_OCTET_STRING`
`Interval`	String	2.5.5.16	65	`ADSTYPE_LARGE_INTEGER`
`LargeInteger`	String	2.5.5.16	65	`ADSTYPE_LARGE_INTEGER`

2.4.2.4.2 Active Directory Syntaxes Not Supported by Oracle Waveset The following table lists the Active Directory syntaxes that are not supported by Oracle Waveset:

Table 2–9 Active Directory Syntaxes Not Supported by Oracle Waveset

Syntax	Syntax ID	OM ID	ADS Type
DN with Unicode string	2.5.5.14	127	ADSTYPE_DN_WITH_STRING
DN with binary	2.5.5.7	127	ADSTYPE_DN_WITH_BINARY
OR-Name	2.5.5.7	127	ADSTYPE_DN_WITH_BINARY
Replica Link	2.5.5.10	127	ADSTYPE_OCTET_STRING
NT Security Descriptor	2.5.5.15	66	ADSTYPE_NT_SECURITY_DESCRIPTOR
Octet String	2.5.5.10	4	ADSTYPE_OCTET_STRING
SID String	2.5.5.17	4	ADSTYPE_OCTET_STRING
UTC Time String	2.5.5.11	23	ADSTYPE_UTC_TIME
Object(Access-Point)	2.5.5.14	127	N/A

Oracle Waveset also supports the `jpegPhoto` and `thumbnailPhoto` account attributes, which use the Replica Link syntax. These attributes are write-only fields. This means that Oracle Waveset does not display the value of these attributes after reconciliation. The `jpegPhoto` and `thumbnailPhoto` attributes can be provisioned only if the account performing the provisioning operation has Admin privileges. Note that the size limit for the `jpegPhoto` and `thumbnailPhoto` attributes is 100 KB, but it is recommended to keep the size below 10K. Similarly, recommended thumbnail photo size in pixels is 96x96. See Adding Byte[] Datatype Attribute to Active Directory Connector Resource Forms for more information about adding these attributes to the user form.

2.4.3 Active Directory Connector Sample Forms

The following sample forms are provided with the Active Directory connector:

- Windows Active Directory Create Container Form (ADcontainercreate.xml)
- Windows Active Directory Create Group Form (ADgroupcreate.xml)
- Windows Active Directory Create Organizational Unit Form (ADorganizationalunitcreate.xml)
- Windows Active Directory Update Container Form (ADcontainerupdate.xml)
- Windows Active Directory Update Group Form (ADgroupupdate.xml)
- Windows Active Directory Update Organizational Unit Form (ADorganizationalunitupdate.xml)

In addition, the following forms are also provided: migration.xml, resourceWizard.xml, postProcess.xml, and userForm.xml.

2.4.4 Resource Object Management

Waveset supports the following Active Directory objects:

Table 2–10 Supported Active Directory Objects

Resource Object	Supported Features	Attributes Managed
Group	Create, update, delete	cn, samAccountName, description, managedby, member, mail, groupType, authOrig, name
DNS Domain	Find	dc
Organizational Unit	Create, delete, find	ou
Container	Create, delete, find	cn, description

The attributes that can be managed on resource objects are also generally dictated by the attribute syntaxes. The attributes for these object types are similar as those for user accounts and are supported accordingly.

2.4.5 Enforcing Check Password History

To check the password history for an Active Directory account when a user changes the password, the user must provide an AD password. To enable this feature, you must pass the current password value to the __CURRENT_PASSWORD__ attribute, and then add this attribute to the End User Change Password form.

1. Go to the Oracle Waveset debug page:

   `http://host_name:port/idm/debug`

2. In the column corresponding to List Objects, select **User Form.**
3. Click **List Objects.**
4. Click the edit button corresponding to the End User Change Password Form.
5. Add the following code snippet:

```
<Field
name='resourceAccounts.currentResourceAccounts[RESOURCE_NAME].attributes.CURREN
T_PASSWORD'>
         <Display class='Text'>
           <Property name='title' value='CurrentPassword'/>
           <Property name='secret' value='true'/>
         </Display>
</Field>
```

6. Click **Save.**
7. Add the __CURRENT_PASSWORD__ attribute as follows:
   a. Click the Resource tab.
   b. Click **Edit Resource Page,** to open the resource to which you want to add the attribute.

      The Edit *RESOURCE_NAME* Resource Wizard page is displayed.

   c. On the Resource Parameters page, click **Next.**
   d. On the Account Attributes page, click **Add Attribute.**
   e. In the new row that is added to the table, specify the following values:
      – In the Identity System User Attributes column, enter CURRENT_PASSWORD in the text field.

- In the Resource User Attribute column, enter `__CURRENT_PASSWORD__`.

f. Click **Save.**

8. Go to the Oracle Waveset debug page:

   `http://host_name:port/idm/debug`

9. Open the resource, search for the CURRENT_PASSWORD attribute, and then add the view as follows:

```
<AccountAttributeType id='60' name='CURRENT_PASSWORD' syntax='encrypted'
mapName='__CURRENT_PASSWORD__' mapType='string' writeOnly='true'>
       <Views>
         <String>Password</String>
         <String>LoginChange</String>
       </Views>
</AccountAttributeType>
```

10. Click **Save.**

From now onward, whenever an attempt to change the account password is made, the user is prompted to enter the current password. The password history is checked before completing the password change operation.

2.5 Troubleshooting the Active Directory Connector

This section provides solutions to problems you might encounter after you deploy or while using the Active Directory connector.

Table 2–11 provides solutions to problems you might encounter with the Microsoft Active Directory User Management connector.

Table 2–11 Troubleshooting the Active Directory Connector

Problem	Solution

The following error is encountered while updating a user:
Account not found in Resource
	This error is encountered if there are multiple domain controllers configured for the domain. To fix this issue, add a field to ResourceWizard.xml as follows:
	`<Field name="resourceAttributes[SyncDomainController].value" required="false">`
	`<Display class="Text">`
	`<Property name="title" value="SyncDomainController"/>`
	`<Property name="help" value="SyncDomainController"/>`
	`</Display>`
	`</Field>`
	
	Reimport the xml file and provide the domain controller (Host) value for the same in the resource form.

3

Oracle Waveset Connector for IBM AS400

This chapter includes the following information about the AS400 connector for Oracle Waveset:

- About the AS400 Connector
- Migrating an AS400 Connector
- Deploying the AS400 Connector
- Using the AS400 Connector
- Troubleshooting the AS400 Connector
- Known Issues for the AS400 Connector

3.1 About the AS400 Connector

- Overview of the AS400 Connector
- Security Considerations for the AS400 Connector
- Certified Components for the AS400 Connector
- Supported Languages for the AS400 Connector

3.1.1 Overview of the AS400 Connector

The AS400 connector for Oracle Waveset supports provisioning to IBM AS400 resources.

The AS400 connector is implemented using the Identity Connector Framework (ICF). The ICF provides a container that separates the connector bundle from the application. The ICF also provides common features that developers would otherwise need to implement on their own, such as connection pooling, buffering, time outs, and filtering. For more information about the ICF, see Chapter 1, "Identity Connectors Overview".

The AS400 connector supersedes the OS/400 resource adapter. To migrate from a resource adapter deployment, see Migrating an AS400 Connector.

This section provides the following information about the AS400 connector:

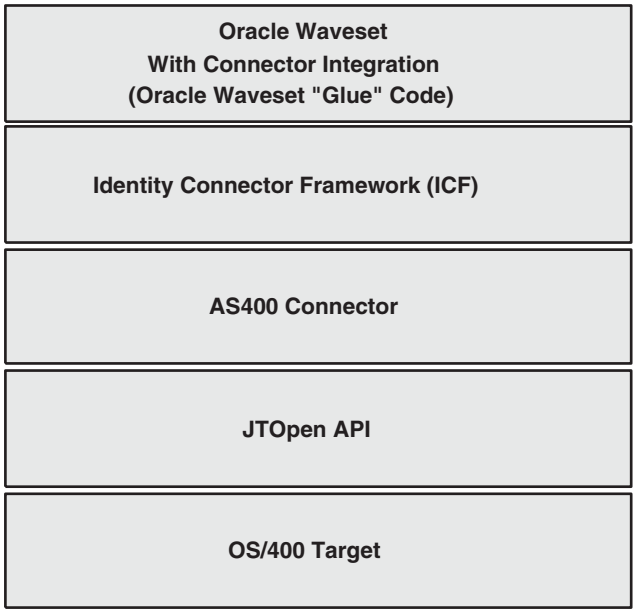
- AS400 Connector Architecture
- AS400 Connector Features
- Configuration Properties for the AS400 Connector
- Resource Object Management for the AS400 Connector

- OS/400 Directory Entry Object Attributes

3.1.1.1 AS400 Connector Architecture

The following figure shows the AS400 connector architecture.

Figure 3–1 AS400 Connector Architecture



The AS400 connector architecture includes these components:

- Oracle Waveset includes the connector integration files. These files are XML files that provide the configuration information necessary to transform data from a resource to Oracle Waveset. Integration files are sometimes called the connector "glue" code.
- The Identity Connector Framework (ICF) provides basic provisioning, logging, and other functions that Oracle Waveset (and Oracle Identity Manager) connectors can use.
- The AS400 connector requires the JTOpen library (`jt400.jar` file) to access the OS/400 target system.

3.1.1.2 AS400 Connector Features

The AS400 connector supports these provisioning operations:

- Create account
- Update account
- Delete account
- Enable/disable account
- Reset password
- Before and after actions

3.1.1.3 Configuration Properties for the AS400 Connector

The AS400 connector for Oracle Waveset supports the configuration parameters shown in the following table.

Table 3–1 Configuration Properties for the AS400 Connector

Name	Type	Required	Description
`adminAccount`	String	Yes	Administrator account name. This property was `adminAcct` for the OS/400 resource adapter.
`adminPassword`	GuardedString	Yes	Administrator password. This property was `password` for the OS/400 resource adapter.
`host`	String	Yes	Hostname or IP address of the AS400 resource to connect to.
`useSSL`	Boolean	Yes	Indicates whether to connect to the host using SSL. The default value is `true`. The `useSSL` property must be set to either `true` or `false`; it cannot be undefined. This property was `ssl` for the OS/400 resource adapter.

3.1.1.4 Resource Object Management for the AS400 Connector

The AS400 connector supports the `__ACCOUNT__` object class, which represents OS/400 user profiles. The AS400 connector can also list OS/400 group profiles, denoted by the `__GROUP__` object class.

3.1.1.5 OS/400 Directory Entry Object Attributes

An AS400 connector resource supports additional new attributes compared to the OS/400 resource adapter, as listed in New Account Attributes for the AS400 Connector.

Those new attributes are stored in the OS/400 directory entry object associated with the account. If these new attributes are not required, you can build your own user form to skip these extra attributes. An alternative approach is to remove those attributes from the Resource Schema using the Resource Wizard.

The AS400 connector creates a directory entry if it is absent on the OS/400 target system, whenever attribute update is invoked on an OS/400 account.

3.1.2 Security Considerations for the AS400 Connector

This section provides the following security information for the AS400 connector:

- Supported Connections for the AS400 Connector
- Required Administrative Privileges for the AS400 Connector

3.1.2.1 Supported Connections for the AS400 Connector

The AS400 connector by default uses Secure Sockets Layer (SSL) to talk to the AS400 resource. The usage of SSL is controlled by the `useSSL` configuration property.

See Configuring SSL for the AS400 Connector.

3.1.2.2 Required Administrative Privileges for the AS400 Connector

> **Note:** The AS400 connector uses an account with the administrative privileges described below. For increased security, it is recommended that you create a separate account, apart from QSECOFR (the OS/400 security officer account).

The following administrative privileges are required for the AS400 connector:

- Create Account — CRT: To add an OS/400 user, the administrator must have all of the following:
  - *SECADM special authority
  - *USE authority to the initial program, initial menu, job description, message queue, output queue, and attention-key-handling program if specified
  - *CHANGE and object management authorities to the group profile and supplemental group profiles, if specified
- Update Account — CHG: The user must have *SECADM special authority, and *OBJMGT and *USE authorities to the user profile being changed, to specify this command. *USE authority to the current library, program, menu, job description, message queue, print device, output queue, or ATTN key handling program is required to specify these parameters.
- Delete Account — DLT: The user must have use (*USE) and object existence (*OBJEXIST) authority to the user profile. The user must have existence, use, and delete authorities to delete a message queue associated with and owned by the user profile. The user profile cannot be deleted if a user is currently running under the profile, or if it owns any objects and OWNOBJOPT(*NODLT) is specified.

  All objects in the user profile must first either be transferred to new owners by using the Change Object Owner (CHGOBJOWN) command or be deleted from the system. This can also be accomplished by specifying OWNOBJOPT(*DLT) to delete the objects or OWNOBJOPT(*CHGOWN user-profile-name) to change the ownership.

  Authority granted to the user does not have to be specifically revoked by the Revoke Object Authority (RVKOBJAUT) command; it is automatically revoked when the user profile is deleted.
- Search or Reconcile Account — DSP: The user name can be specified as USRPRF(*ALL) or USRPRF(generic*-user-name) only when TYPE(*BASIC) and OUTPUT(*OUTFILE) are specified.

Note: If the administrator requires additional rights, use the following commands from the OS/400 console:

```
CRTUSRPRF USRPRF (adminUserName) AUT
(list-of-necessary-permissions)
CHGUSRPRF USRPRF (adminUserName) SPCAUT
(list-of-necessary-permissions)
```

The *list-of-necessary-permissions* can differ for each administrator and should be determined based on your deployment's requirements.

Also, *USE and *CHANGE are values for the GRPAUT (Group Authority) parameter of the CHGUSRPRF command. Group Authority specifies the authority given to the group profile for newly created objects.

3.1.3 Certified Components for the AS400 Connector

The AS400 connector for Oracle Waveset is certified with the following components:

Table 3–2 Certified Components for the AS400 Connector

Component	Requirement
Oracle Waveset	Oracle Waveset 8.1.1 Patch 4 or later
Target Systems	OS/400 v5r4, IBM i 6.1, and IBM i 7.1
JTOpen library (`jt400.jar`)	6.2
JDK	JDK 1.5 or later

3.1.4 Supported Languages for the AS400 Connector

The AS400 connector is localized in the following languages:

- Arabic
- Chinese (Simplified)
- Chinese (Traditional)
- Danish
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazilian)
- Spanish

To change the language, specify the `lang` and `cntry` URL parameters. For example, to access Oracle Waveset in the Japanese language:

```
http://domain:port/idm/?lang=ja&cntry=JP
```

3.2 Migrating an AS400 Connector

If you currently have the OS/400 resource adapter installed, this section describes how to migrate to the AS400 connector.

Oracle Waveset provides the required connector glue code. The installation script places the connector bundle into the proper locations on the application server and loads the required upgrade XML files.

3.2.1 Migrating from an OS/400 Resource Adapter

To migrate to an AS400 connector, follow these steps:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the AS400 Connector.
2. Log in to the Oracle Waveset Administrator interface as an Administrator.
3. Select the Resources tab and then the Migrate Adapters tab.
4. Follow the Migration Wizard to complete the migration. A script runs in the background that updates the schema map.

3.2.2 Updating the Schema Map

The AS400 connector is backwards compatible with the OS/400 resource adapter. All forms, workflows, and tasks should work the same as before you migrate the adapter.

- New Account Attributes for the AS400 Connector
- Renamed Account Attributes for the AS400 Connector
- New Operation Options

3.2.2.1 New Account Attributes for the AS400 Connector

The AS400 connector supports the following new account attributes. These attributes were not supported by default in the OS/400 resource adapter.

Note: These new attributes require a directory entry object to be created on the OS/400 target system. The directory entry has lazy initialization, so it is created only if the application explicitly provides any of these new attributes.

For a description of these new attributes, including the specific values allowed, see Table 3–4, " Account Attributes for the AS400 Connector".

- `PASSWORD_CHANGE_INTERVAL`
- `ACCOUNTING_CODE`
- `ADDRESS1`
- `ADDRESS2`
- `BUILDING`
- `COMPANY`
- `DEPARTMENT`
- `FAX`

- FIRST_NAME
- FULL_NAME
- GROUP_AUTHORITY
- GROUP_PROFILE_NAME
- JOB_TITLE
- LAST_NAME
- LOCATION
- MIDDLE_NAME
- OFFICE
- PREFERRED_NAME
- STORAGE_USED
- SUPGRPPRF
- TELEPHONE

3.2.2.2 Renamed Account Attributes for the AS400 Connector

The following account attributes have been renamed for the AS400 connector:

Table 3–3 Renamed Account Attributes for the AS400 Connector

Adapter Attribute Name	Connector Attribute Name	Connector Data Type
accountId	__NAME__	String
CHANGE_DATE	__LAST_PASSWORD_CHANGE_DATE__	Long
expirePassword	__PASSWORD_EXPIRED__	Boolean
password	__PASSWORD__	GuardedString
PREVIOUS_SIGN_ON	__LAST_LOGIN_DATE__	Long
STATUS	__ENABLE__	Boolean

3.2.2.3 New Operation Options

The legacy OS/400 resource adapter supported a customized version of the Delete user operation, supported by the `OS400Deprovision` form included in the sample directory of the Waveset installation. The AS400 connector preserves this support, although the following account attributes used for this feature have been replaced with operation options:

- GRPPRF
- OWNOBJOPT

3.3 Deploying the AS400 Connector

You can deploy the AS400 connector either locally in Oracle Waveset or remotely in the Connector Server, as described in the following sections:

- Installing the AS400 Connector in Oracle Waveset
- Installing the AS400 Connector in the Connector Server

Note: In a production environment, it is recommended that you deploy the AS400 connector in the Connector Server.

3.3.1 Installing the AS400 Connector in Oracle Waveset

To install the AS400 connector, you must have access to the file system on the application server.

In the following procedure, *WavesetInstallDir* refers to the location where Oracle Waveset is deployed.

To install the AS400 connector in Oracle Waveset, follow these steps:

1. Download the JTOpen library from the following location:

   http://jt400.sourceforge.net

2. Unzip the JTOpen library in a temporary directory and find the `jt400.jar` file.
3. Stop the Oracle Waveset web application.
4. If a previous version of the`jt400.jar` file does not already exist, copy the `jt400.jar` file to the *WavesetInstallDir*`/WEB-INF/lib` directory.

   or

   If a previous version of the`jt400.jar` file does already exist, copy the `jt400.jar` file inside the AS400 connector JAR, as follows:

   a. Go to the *WavesetInstallDir*`/WEB-INF/bundles` directory, where the `org.identityconnectors.as400` bundle resides.

   b. Add the `jt400.jar` file to the `/lib` subdirectory of the AS400 connector JAR. The result is the following layout inside the AS400 connector JAR:

   ```
   org.identityconnectors.as400-1.0.0.jar
     /META-INF
     /org
     /lib
       jt400.jar
   ```

5. Start the Oracle Waveset web application.
6. Log in to the Oracle Waveset Administrator interface.

3.3.2 Installing the AS400 Connector in the Connector Server

Before you begin, consider these requirements. For the JDK requirements. see Certified Components for the AS400 Connector. If necessary, see your `JAVA_HOME` environment variable to point to your specific installation.

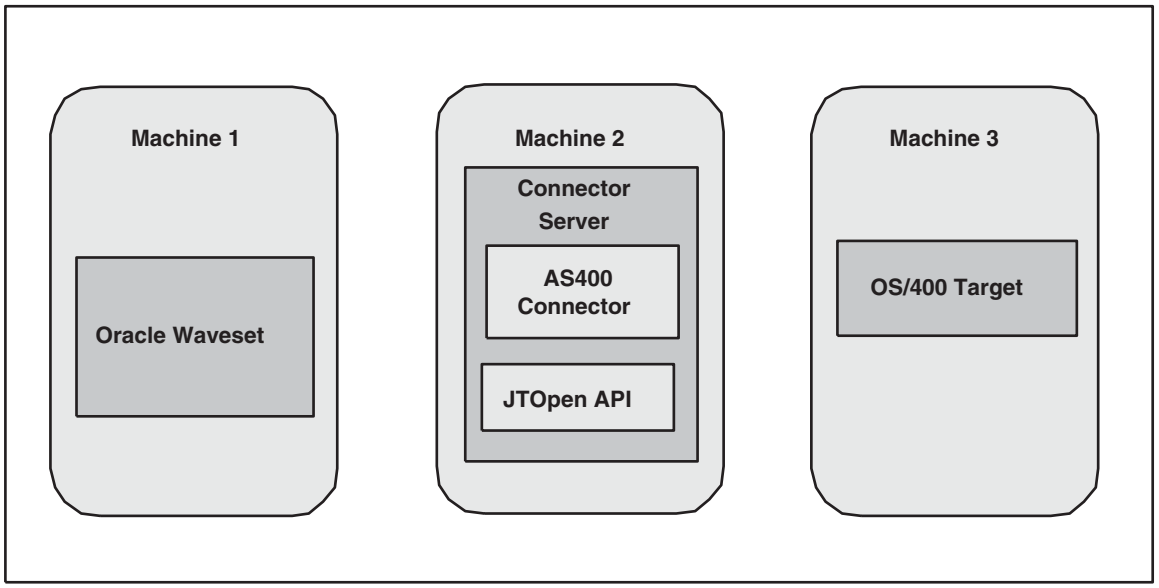
This section describes the following subsections:

- AS400 Connector Deployment Architecture With the Connector Server
- Installing and Configuring the Connector Server
- Running the Connector Server on Windows Systems
- Running the Connector Server on UNIX and Linux Systems
- Installing the AS400 Connector in the Connector Server

3.3.2.1 AS400 Connector Deployment Architecture With the Connector Server

If you install the AS400 connector in the Connector Server, the following figure shows the distributed deployment architecture.

Figure 3–2 AS400 Connector Deployment Architecture With the Connector Server



- **Machine 1** has Oracle Waveset deployed.
- **Machine 2** has the AS400 connector installed in the Connector Server. The Connector Server is part of the Identity Connector Framework (ICF).

  The `jt400.jar` file from the JTOpen library must be installed in the `CONNECTOR_SERVER_HOME/lib` directory.

  For detailed installation information, see Installing the AS400 Connector in the Connector Server.
- **Machine 3** has the OS/400 target deployed.

3.3.2.2 Installing and Configuring the Connector Server

To install and configure the Connector Server, follow these steps:

1. Create a new directory on the machine where you want to install the Connector Server. In this section, `CONNECTOR_SERVER_HOME` represents this directory.
2. Unzip the Connector Server package in your new directory from Step 1. The Connector Server package is available with the Identity Connector Framework (ICF).
3. In the `ConnectorServer.properties` file, set the following properties, as required by your deployment. The `ConnectorServer.properties` file is located in the `conf` directory.

Property	Description
`connectorserver.port`	Port on which the Connector Server listens for requests. The default is 8759.

Property	Description
connectorserver.bundleDir	Directory where the connector bundles are deployed. The default is bundles.
connectorserver.libDir	Directory in which to place dependent libraries. The default is lib.
connectorserver.usessl	If set to true, the Connector Server uses SSL for secure communication. The default is false. If you specify true, use the following options on the command line when you start the Connector Server: ■ -Djavax.net.ssl.keyStore ■ -Djavax.net.ssl.keyStoreType (optional) ■ -Djavax.net.ssl.keyStorePassword
connectorserver.ifaddress	Bind address. To set this property, uncomment it in the file (if necessary). The bind address can be useful if there are more NICs installed on the machine.
connectorserver.key	Connector Server key.

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

     For more information, see Running the Connector Server on Windows Systems or Running the Connector Server on UNIX and Linux Systems.
   - For all other properties, edit the ConnectorServer.properties file manually.
5. The conf directory also contains the logging.properties file, which you can edit if required by your deployment.

3.3.2.3 Running the Connector Server on Windows Systems

To run the Connector Server on Windows systems, use the ConnectorServer.bat script, as follows:

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

   The ConnectorServer.bat script supports the following options:

Option	Description
/install [*serviceName*] ["-J *java option*"]	Installs the Connector Server as a Windows service. Optionally, you can specify a service name and Java options. If you do not specify a service name, the default name is ConnectorServerJava.

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

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

3.3.2.4 Running the Connector Server on UNIX and Linux Systems

To run the Connector Server on UNIX and Linux systems, use the `connectorserver.sh` script, as follows:

1. Make sure that you have set the properties required by your deployment in the `ConnectorServer.properties` file, as described in Installing and Configuring the Connector Server.
2. Change to the `CONNECTOR_SERVER_HOME/bin` directory.
3. Use the `chmod` command to set the permissions to make the `connectorserver.sh` script executable.
4. Run the `connectorserver.sh` script. The script supports the following options:

Option	Description
`/run [ -Jjava-option ]`	Runs the Connector Server in the console. Optionally, you can specify one or more Java options. For example, to run the Connector Server with SSL: `./connectorserver.sh /run -J-Djavax.net.ssl.keyStore=mykeystore.jks -J-Djavax.net.ssl.keyStorePassword=password`
`/start [ -Jjava-option ]`	Runs the Connector Server in the background. Optionally, you can specify one or more Java options.
`/stop`	Stops the Connector Server, waiting up to 5 seconds for the process to end.
`/stop n`	Stops the Connector Server, waiting up to *n* seconds for the process to end.
`/stop -force`	Stops the Connector Server. Waits up to 5 seconds and then uses the `kill -KILL` command, if the process is still running.
`/stop n -force`	Stops the Connector Server. Waits up to *n* seconds and then uses the `kill -KILL` command, if the process is still running.

Option	Description
`/setKey key`	Sets the Connector Server key. The `connectorserver.sh` script stores the hashed value of *key* in the `connectorserver.key` property in the `ConnectorServer.properties` file.

3.3.2.5 Installing the AS400 Connector in the Connector Server

To install the AS400 connector for Oracle Waveset into the Connector Server, follow these steps:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the AS400 Connector.
2. Stop the Connector Server.
3. Copy the AS400 connector bundle into the Connector Server `CONNECTOR_SERVER_HOME/bundles` directory.
4. Copy the `jt400.jar` file to the `CONNECTOR_SERVER_HOME/lib` directory.
5. Start the Connector Server.

For information about starting and stopping the Connector Server, see Running the Connector Server on Windows Systems or Running the Connector Server on UNIX and Linux Systems.

3.4 Using the AS400 Connector

- OS/400 Objects Associated with an Account on an OS/400 Resource
- Special Characters in Passwords
- Account Attributes for the AS400 Connector
- Sample Forms for the AS400 Connector
- Before and After Actions for the AS400 Connector
- Connection Pooling for the AS400 Connector
- Configuring SSL for the AS400 Connector

3.4.1 OS/400 Objects Associated with an Account on an OS/400 Resource

Oracle Waveset supports three options for handling OS/400 objects that are associated with an account on an OS/400 resource. To enable this specialized support, you must use the `OS400Deprovision` form, which is located in the Oracle Waveset sample directory.

You must also edit the system configuration object. The instructions for editing this object are included in the comments in the `OS400Deprovision` form. Once enabled, these options appear on the Delete Resource Accounts page when you choose to delete a user's OS/400 resource account.

The form field OWNOBJOPT can have one of the following values:

- DLT. The user's resource account and associated objects are deleted.
- NODLT. The dependent objects are reassigned to a default profile (QDEFOWN) instead of being deleted.
- CHGOWN. The user's dependent objects are inherited by a specified custom OS/400 profile, which is specified by the value of the GRPPRF field.

The form field GRPPRF is the user profile name that inherits objects owned by the deleted OS/400 account. This field is optional. It is relevant only when the field OWNOBJOPT has the value of CHGOWN.

3.4.2 Special Characters in Passwords

The AS400 connector supports special characters in passwords for OS/400 version 5.1 or later.

A password must begin with an uppercase alphabetic character (A-Z) or the special characters `@`, `$`, `#`, or `_`, followed by uppercase alphabetic characters A-Z, numbers 0-9, and the special characters `@`, `$`, `#`, and `_`. The maximum length of a password is 10 characters.

3.4.3 Account Attributes for the AS400 Connector

The AS400 connector attributes naming follows the standard OS/400 conventions. For more information about OS/400 attributes, see the Create User Profile (`CRTUSRPRF`) and Change Directory Entry (`CHGDIRE`) OS/400 commands at the following site:

`http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp`

The following table describes the account attributes for the AS400 connector. All attributes are strings, unless indicated otherwise. Also, unless indicated, the default properties for each attribute are not required, creatable, updatable, readable, and returned by default.

Table 3–4 Account Attributes for the AS400 Connector

AS400 Connector Attribute	Native OS/400 Attribute	Description
`__ENABLE__`	None	Boolean. Indicates whether the account is enabled and logins are allowed. This attribute was `STATUS` for the OS/400 adapter.
`__LAST_LOGIN_DATE__`	None	Long. Read-only. Last login date. This attribute was `PREVIOUS_SIGN_ON` for the OS/400 adapter.
`__LAST_PASSWORD_CHANGE_DATE__`	None	Long. Read-only. Date and time the password was last updated. This attribute was `CHANGE_DATE` for the OS/400 adapter.
`__NAME__`	User profile name	Required. Not updatable. OS/400 user profile name. This attribute was `accountId` for the OS/400 adapter. The user profile name can be a maximum of 10 characters, including any letter (A-Z), a number (0-9), and the following special characters: pound (`#`), dollar (`$`), underscore (`_`), and at (`@`). The first character cannot be a number.
`__PASSWORD__`	User password	Required. Guarded string. OS/400 user password. Value is encrypted. This attribute was `password` on the OS/400 adapter.

Table 3–4 (Cont.) Account Attributes for the AS400 Connector

AS400 Connector Attribute	Native OS/400 Attribute	Description
PASSWORD_CHANGE_INTERVAL	None	Integer. Number of days between the date when the password is changed and the date when the password expires. Values can be -1 through 366: ■ -1 - The user's password does not expire (*NOMAX). ■ 0 - The system value QPWDEXPITV is used to determine the user's password expiration interval (*SYSVAL). ■ 1-366 days.
__PASSWORD_EXPIRED__	None	Boolean. Indicates whether the password has expired. This attribute was expirePassword for the OS/400 adapter.
ACCOUNTING_CODE	ACGCDE	Accounting code associated with the user. Values can be a character value (15 characters, padded with blanks if fewer that 15 characters), *SAME, or *BLANK.
ADDRESS1	Directory entry attribute	First line of the user's address.
ADDRESS2	Directory entry attribute	Second line of the user's address.
ASTLVL	ASTLVL	Assistance level. Sets which interface to use.
ATNPGM	ATNPGM	Attention-key-handling program for this user.
BUILDING	Directory entry attribute	Building name or number.
CCSID	CCSID	Coded character set identifier.
CNTRYID	CNTRYID	Country or region identifier.
COMPANY	Directory entry attribute	Company name.
CURLIB	CURLIB	Current library for jobs initiated by this user profile.
DAYS_UNTIL_PASSWORD_EXPIRES	None	Integer. Read-only. Number of days until the password expires.
DEPARTMENT	Directory entry attribute	Department name or code.
DLVRY	DLVRY	Delivery mode that specifies how messages sent to the message queue for this user are to be delivered.
FAX	Directory entry attribute	Fax telephone number.
FIRST_NAME	Directory entry attribute	User's first name. A maximum of 20 characters is allowed.
FULL_NAME	Directory entry attribute	User's full name.
GID	GID	Long. Group identification number for this user profile. You can assign the GID to a user who does not have an associated group profile.
GROUP_AUTHORITY	GRPAUT	Authority given to the group profile for newly created objects. Values can be *SAME, *NONE, *ALL, *CHANGE, *USE, or *EXCLUDE.

Table 3–4 (Cont.) Account Attributes for the AS400 Connector

AS400 Connector Attribute	Native OS/400 Attribute	Description
GROUP_PROFILE_NAME	GRPPRF	User's group profile name whose authority is used if no specific authority is given for the user or *NONE.
HIGHEST_SCHEDULING_PRIORITY	PTYLMT	Integer. highest scheduling priority the user is allowed to have for each job submitted to the system. Values can be 0 (highest) through 9 (lowest).
HOMEDIR	HOMEDIR	Pathname of the user's home directory.
INLMNU	INLMNU	Initial menu displayed when the user signs on the system if the user's routing program is the command processor.
INLPGM	INLPGM	For an interactive job, the program called whenever a new routing step is started that has QCMD as the request processing program.
JOBD	JOBD	Fully qualified integrated file-system path name of the job description used for jobs that start through subsystem work station entries.
JOB_TITLE	Directory entry attribute	Job title for this user.
KBDBUF	KBDBUF	Keyboard buffering used when a job is initiated for this user.
LANGID	LANGID	Language identifier for the user.
LAST_NAME	Directory entry attribute	User's last name. A maximum of 40 characters is allowed.
LMTCPB	LMTCPB	Limit capabilities for this user.
LMTDEVSSN	LMTDEVSSN	Limit for number of device sessions for this user.
LOCATION	Directory entry attribute	Location for this user.
MAXSTG	MAXSTG	Maximum amount of auxiliary storage in kilobytes. The value *NOMAX on the OS/400 target system is mapped to -1 on Oracle Waveset.
MIDDLE_NAME	Directory entry attribute	User's middle name.
MSGQ	MSGQ	Message queue where messages are sent for this user.
OFFICE	Directory entry attribute	Office name or number.
OUTQ	OUTQ	Output queue for this user profile.
OWNER	OWNER	Owner of new objects created by this user.
PREFERRED_NAME	Directory entry attribute	User's preferred name.
PRTDEV	PRTDEV	Default print device for this user.
SIGN_ON_ATTEMPTS_NOT_VALID	None	Integer. Read-only. Number of invalid login attempts since the last successful login.
SPCAUT	SPCAUT	List of special authorities for this user. Can have multiple values.
SPCENV	SPCENV	Special environment for this user.

Table 3–4 (Cont.) Account Attributes for the AS400 Connector

AS400 Connector Attribute	Native OS/400 Attribute	Description
SRTSEQ	SRTSEQ	Sort sequence table used for string comparisons for this user.
STORAGE_USED	None	Integer. Read-only. Amount of auxiliary storage in kilobytes occupied by this user's owned objects. Default is 12 kilobytes.
SUPGRPPRF	SUPGRPPRF	List of the user's supplemental group profiles. Can have multiple values. **Note**. To update the supplemental group (SUPGRPPRF) attribute, the group profile attribute must have a non-empty value. That is, to populate supplemental groups, a primary group (GRPPRF attribute) must already be defined.
TELEPHONE	Directory entry attribute	Telephone number.
TEXT	TEXT	Text up to 40 characters describing the object (OS/400 account).
UID	UID	Long. User identification number that identifies a user on the OS/400 target system. Range is 1 to 4294967294. The UID must not already be assigned to another user profile. **Note**. The UID is read-only (that is, non creatable and non-updatable). See also Bug 11671704: UID Attribute is Read-Only.
USRCLS	USRCLS	Type of user associated with this user profile: security officer, security administrator, programmer, system operator, or user.
USROPT	USROPT	Level of help information detail to be shown and the function of the Page Up and Page Down keys by default.

Note: The OWNOBJOPT (owned object option) attribute and OWNOBJOPT (group profile) attribute have been superseded by the respective OS/400 operation.

3.4.4 Sample Forms for the AS400 Connector

The following forms are supplied with the AS400 connector:

- OS400UserForm.xml provides a more user-friendly OS/400 form.
- OS400Deprovision.xml enables more refined control for a delete user operation.

3.4.5 Before and After Actions for the AS400 Connector

The AS400 connector supports Before and After actions written in the OS/400 Command Language. The following example shows a sample After action that sets the value of the TEXT attribute to the text specified by 'new text description' for a recently created account:

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<!DOCTYPE Waveset PUBLIC 'waveset.dtd' 'waveset.dtd'>
<Waveset>
  <ResourceAction name='AfterCreateOS400'>
    <ResTypeAction restype='OS/400' timeout='7000'
     execMode='resource' actionType='OS/400 CL'>
      <act>
      CHGUSRPRF USRPRF($__NAME__$) TEXT('new text description')
      </act>
    </ResTypeAction>
  </ResourceAction>
</Waveset>
```

For more information, see Chapter 51, "Adding Actions to Resources" in the *Oracle Waveset 8.1.1 Resources Reference*.

3.4.6 Connection Pooling for the AS400 Connector

Connection pooling involves the management of AS400 connector instances, so that an OS/400 connection does not have to be created each time an operation is executed. For most applications, the default connection pooling setup should be sufficient. However, the fine-tuning of connection pooling can help to increase throughput, if maximum performance is a concern.

The AS400 connector uses Identity Connector Framework (ICF) connection pooling. For more information, see Editing Connection Pooling Parameters.

3.4.7 Configuring SSL for the AS400 Connector

This section describes how to configure Secure Sockets Layer (SSL) for the AS400 connector. In summary, you must fetch the SSL certificate from the OS/400 resource and then import the certificate in the application server you are using.

Before you begin, consider these requirements:

- For the JDK requirements. see Certified Components for the AS400 Connector. If necessary, set your `JAVA_HOME` environment variable to point to your specific installation.
- SSL must be enabled on the OS/400 server, and the Digital Certificate Manager must be started. For more information, refer to the following document:

  `http://www-912.ibm.com/s_dir/slkbase.NSF/DocNumber/28604514`

To configure SSL for the AS400 connector, follow these steps:

1. Fetch the certificate from the target OS/400 system:
   a. In a web browser, go to the Digital Certificate Manager on `http://`*`OS400domain`*`:2001`, where *`OS400domain`* is the target OS/400 system. Use the same user account and password that you use to access the target OS/400 system.
   b. In the left panel, select Create Certificate Authority.

      Or, if the Create Certificate Authority is not an option, select Install Local CA Certificate on Your PC.
   c. Select Install Certificate, and copy the certificate to a text file. For example: `cert.txt`
2. Determine the SSL keystore location on the application server you are using.

   For example, for Oracle WebLogic Server:

a. Open the WebLogic Server Administration Console (`http://WeblogicDomain:port/console`).

b. Look for SSL configuration settings and specifically the name of the keystore. Sometimes, you will see the full path to the keystore, but other times you will see a name such as "DemoTrust" keystore with a path such as `WEBLOGIC_HOME/server/lib/DemoTrust.jks`.

c. Use the `keytool -importcert` command to add the certificate from Step 1 to the keystore for the specific application server. For example, for WebLogic Server:

```
keytool -importcert -file path-to-certificate -alias arbitrary-alias
-keystore WEBLOGIC_HOME/server/lib/DemoTrust.jks
```

where:

– *`path-to-certificate`* is the path to the certificate file you obtained in Step 1.

– *`arbitrary-alias`* is a user-defined alias for identification of the certificate in the certificate store.

3. To verify presence of the certificate in the certificate store, use the `keytool -list -keystore` command.

3.4.7.1 Using SSL to Communicate with a Connector Server

Follow these steps to communicate with a Connector Server using SSL:

1. Deploy an SSL certificate to the Connector Server's system.
2. Configure your Connector Server to provide SSL sockets.
3. Configure your application to communicate with the Connector Server using SSL.

Refer to the target system's manual for specific information about configuring connections to identity Connector Servers. You should indicate to your application that an SSL connection is required when establishing a connection for each SSL-enabled Connector Server. Additionally, if any of the SSL certificates used by your Connector Servers are issued by a non-standard certificate authority, your application must be configured to respect the additional authorities. Refer to your target system's manual for notes regarding certificate authorities.

Note: Java applications can solve the issue of non-standard certificate authorities by expecting the following Java system properties to be passed when launching the application:

- `javax.net.ssl.trustStorePassword`

  For example:
  `-Djavax.net.ssl.trustStorePassword=changeit`

- `javax.net.ssl.trustStore`

  For example:
  `-Djavax.net.ssl.trustStore=/usr/myApp_cacerts`

Alternately, non-standard certificate authorities can be imported to the standard `${JAVA_HOME}/lib/security/cacerts` directory.

3.5 Troubleshooting the AS400 Connector

Use the Oracle Waveset debug pages to set trace options on the following class:

```
org.identityconnectors.as400.AS400Connector
```

This class returns all available error messages from the OS/400 resource.

3.6 Known Issues for the AS400 Connector

3.6.1 Bug 11671704: UID Attribute is Read-Only

The UID attribute is a unique number that identifies a user on the OS/400 target system. In Oracle Waveset, the UID attribute is part of the default user form. However, if you try to update the UID attribute in Oracle Waveset, an error message is displayed.

Workaround. To update the UID attribute, use the OS/400 Command Language (CHGUSRPRF command).

3.6.2 Bug 12636537: Multi-Valued SPCAUT Attribute Does Not Allow Adding Multiple Values

The SPCAUT attribute is multi-valued, but the user form does not have an Add/Remove button like other multi-valued attributes.

Workaround. Use a space to delimit multiple values for the SPCAUT attribute. For example, in order to set the security administrator and auditor special authorities, specify: *SECADM *AUDIT.

You must explicitly include all the required special authorities, including previous special authorities you want to preserve.

3.6.3 Bug 12635601: Provisioning User with Values for MSGQ and JOBD Fails

This problem occurs as follows:

1. Create an AS400 connector resource in Oracle Waveset.
2. Create an Oracle Waveset user.
3. Create an OS/400 account for the Oracle Waveset user from Step 2 with the JOBD (job description) attribute set to the value QGPL/QDFTJOBD.

Oracle Waveset returns an error. This problem can occur for these attributes: INLMNU, INLPGM, JOBD, MSGQ, ATNPGM, OUTQ, and SRTSEQ. This problem also occurs if you assign an Oracle Waveset user to multiple OS/400 accounts on different OS/400 hosts.

Workaround. When you create the OS/400 account for the Oracle Waveset user, specify the fully-qualified path for these attributes. For example, for the job description attribute:

```
/QSYS.LIB/QGPL.LIB/QDFTJOBD.JOBD
```

4

Oracle Waveset Connector for IBM Domino

This chapter includes the following information about the Domino connector for Oracle Waveset:

4.1 About the Domino Connector

4.1.1 Overview of the Domino Connector

The Domino connector for Oracle Waveset supports provisioning to IBM Domino servers. The Domino connector supersedes the Domino resource adapter and previous versions of the Domino connector. For migration information, see Migrating to the Domino Connector.

The Domino connector is implemented using the Identity Connector Framework (ICF). The ICF provides a container that separates the connector bundle from the application. The ICF also provides common features that developers would otherwise need to implement on their own, such as connection pooling, buffering, time outs, and filtering. For more information about the ICF, see Chapter 1, "Identity Connectors Overview".

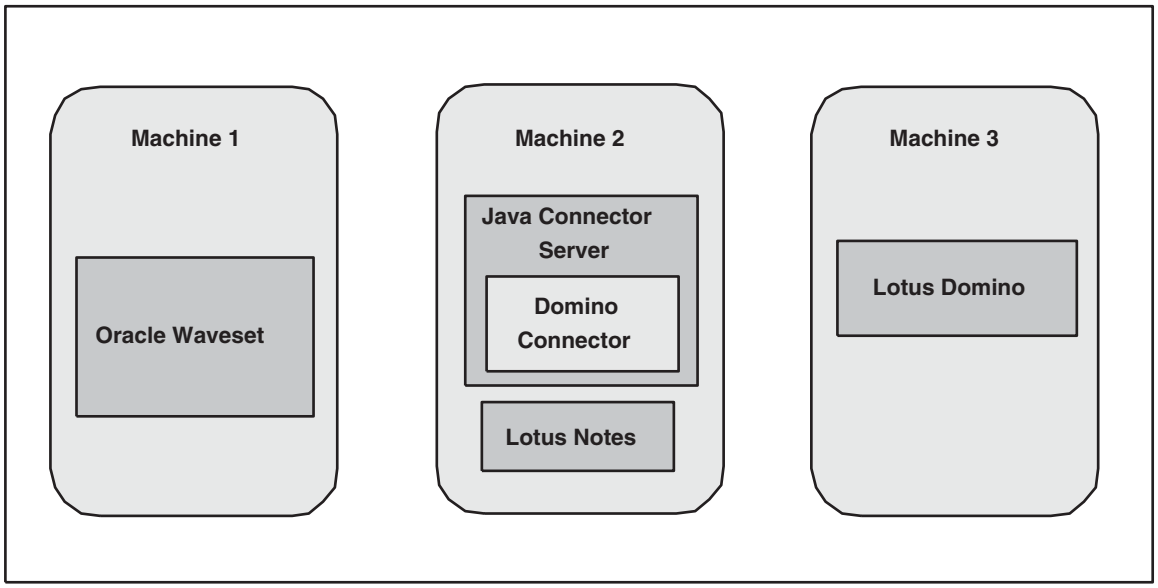
This section provides the following additional information about the Domino connector:

- Resource Object Management

4.1.1.1 Domino Connector Deployment Architecture

The Domino connector requires the distributed deployment architecture, as shown in the following figure.

Figure 4–1 Domino Connector Deployment Architecture



- **Machine 1** has Oracle Waveset deployed.

  Note. Deploying the Domino connector on the Oracle Waveset machine is not certified. You must deploy the Domino connector in the Java Connector Server (Machine 2 in the figure).

- **Machine 2** has the Domino connector deployed. This machine must meet the following requirements:

  - The Java Connector Server is deployed on this machine. The Domino connector is deployed in the Java Connector Server.

    The Java Connector Server is part of the Identity Connector Framework (ICF). The Domino connector requires ICF version 1.0 or later.

  - The Lotus Notes client (32-bit version) must be installed on the same machine where the Domino connector is deployed. Also:

    – The `Notes.jar` file (included with Lotus Notes) must be available to the Domino connector.

    – The `PATH` variable must specify the directory where `nnotes.dll` resides.

  - Machine 2 must be running Windows Server 2003, Windows Server 2008, or a desktop machine (including Windows 7, Windows Vista, and Windows XP).

    Both the Windows 32-bit and 64-bit platforms are supported. However, the Domino connector uses the 32-bit Domino C API and therefore is supported only with the 32-bit version of Lotus Notes.

    You must install the Lotus Notes client on the Windows machine where the Domino connector is running.

- **Machine 3** has the Lotus Domino target deployed.

  Machine 3 can be running Windows Server 2003 or Windows Server 2008, 32-bit and 64-bit versions, as well as both the 32-bit and 64-bit versions of Lotus Domino.

4.1.1.2 Domino Connector Features

The Domino connector for Oracle Waveset supports these provisioning operations:

- Authentication
- Create
- Delete
- Get
- Schema
- Before and After actions. Supported types are:
  - Lotus Script (`ActionType` is `lotusscript`)
  - `cmd` shell (`ActionType` is `cmd`)

  For more information see, Executing Before and After Actions.
- Search
- Sync
- Test
- Update
- Validate
- Group CRUD. Domino resource adapter group forms can be reused.

4.1.1.3 Domino Connector Configuration Parameters

The Domino connector for Oracle Waveset supports the configuration parameters shown in the following table.

Table 4–1 Domino Connector Configuration Parameters

Name	Type	Required	Description
`Admin ID File`	String	Yes	Fully-qualified path to the Administrator ID file. Default: `C:\Lotus\Notes\Data\admin.id`
`Administration Server`	String	Yes	Name of the host where the administration server is running.
`Admin Account name`	String	Yes	Administrator account name, such as `Administrator/ACM`.
`Admin Password`	GuardedString	Yes	Password of the administrator.
`Certifier ID File`	String	Yes	Fully-qualified path to the Certifier ID file. Default: `C:\Lotus\Domino\Data\cert.id`
`Certifier ID Password`	GuardedString	Yes	Password for the specified Certifier ID file.

Table 4–1 (Cont.) Domino Connector Configuration Parameters

Name	Type	Required	Description
`Create ID File`	Boolean	No	Indicates if an ID File should be created for this account. Default: `TRUE`
`Create Mail DB`	Boolean	No	Indicates whether to set up mail when a user is created. If checked (`TRUE`), mail setup occurs at account creation. If unchecked (`FALSE`), mail setup occurs at first login. Default: `TRUE`
`Default Password Expiry`	Integer	No	Default password expiry time, in days. Default: 720
`ID Type`	Integer	No	Type of ID file: 0 for flat or 1 for hierarchical. Default: 1
`Is North American`	Boolean	No	Check (`TRUE`) if the server is in North America, or uncheck (`FALSE`) if not. Default: `TRUE`
`Mailfile removal Option`	Integer	Yes	Values can be: 0 = Preserve mail file. 1 = Delete just mail file specified in person record. 2 = Delete mail file specified in person record and all replicas.
`Mail Server`	String	No	Default mail server to use when creating users. Use the abbreviated format. For example: `server/org`
`Default Mail System`	Integer	No	Indicates the default mail system when creating users: 0 = Notes 1 = CCMAIL 2 = VIMMail 99 = None
`Minimum Password Strength`	Integer	No	Defines the password strength required for subsequent changes to the password by the user. A password assigned in the methods listed above is not initially checked against the strength. Domino measures a password's strength and security according to the level assigned on its password quality scale. Default: 6 The scale ranges from Weak to Strong or from 0 (lowest - no password required) to 16 (highest).

Table 4–1 (Cont.) Domino Connector Configuration Parameters

Name	Type	Required	Description
Notes ini file	String	Yes	Full path to the Lotus Notes initialization file, including the file name. Default: C:\Lotus\Notes\notes.ini
Notes installation folder	String	Yes	Location where the client is installed and where the notes.exe file is located. Default: C:\Lotus\Notes
Explicit Policy Name	String	No	Name of the Domino explicit policy to be assigned to the user. When set, this value could modify or override other user attribute values. Refer to the Domino documentation for more information.
Certification Log	String	No	Fully qualified path to the certification log residing on the server machine. Default: C:\Lotus\Domino\Data\certlog.nsf
Registration Server Machine	String	Yes	Name of the Domino server to use when registering new Domino accounts.
Roaming Cleanup	Integer	No	Cleanup setting for files belonging to roaming Domino accounts. Values can be: 0 = Never 1 = Periodically in days 2 = At shutdown 3 = Prompt
Roaming Cleanup Period	Integer	No	If the value of Roaming Cleanup is 1, specifies the period in days that cleanup will be performed.
Roaming Replica Servers	String	No	List of servers that will contain replicas of roaming files.
Roaming Server	String	No	Server destination for roaming files belonging to a Domino account.
Store ID In AddrBook	Boolean	No	Check (TRUE) if the ID should be stored in the Domino Address Book; otherwise, uncheck (FALSE). Default: TRUE
Update Addr Book	Boolean	No	Check (TRUE) if the Address Book should be updated using the new ID file; otherwise, uncheck (FALSE).
Store ID In A Mail File	Boolean	No	Check (TRUE) if the ID should be stored in a file; otherwise, uncheck (FALSE). Default: FALSE
Filename of Names Database	String	Yes	Filename of the Names Database. Default: names.nsf

Table 4–1 (Cont.) Domino Connector Configuration Parameters

Name	Type	Required	Description
`Sync Internet Password`	Boolean	No	Set to True to synchronize the user's Internet password. The Internet password is in the user's Person document in the Domino Directory. If the user changes the password for the Notes client ID, the Internet password automatically (but not immediately) changes to match it. Default: False
`Use ID Vault`	Boolean	No	If set to True, the ID Vault's password reset feature is used when changing the user's password. Default: False

4.1.1.4 Resource Object Management

The Domino connector manages the following native Domino objects.

Table 4–2 Native Domino Objects

Resource Object	Supported Features	Attributes Managed
Group	create, delete, list, rename, saveas, update	ConflictAction, Group_Main, AvailableForDirSync, DeleteNTUserAccount, DocumentAccess, Form, GroupName, GroupTitle, GroupType, InternetAddress, ListCategory, ListDescription, ListName, ListOwner, LocalAdmin, MailDomain, MailVerify, Owner, Type, Members, MemberPeople, MemberGroups

4.1.2 Domino Connector Requirements

The Domino connector for Oracle Waveset has the following requirements:

- The Domino connector uses the 32-bit Domino C API and can run on Microsoft Windows 32-bit and 64-bit platforms. However, the Domino connector runs with the 32-bit Lotus Notes and Lotus Domino only.
- The Lotus Notes client must be installed on the Windows machine where the Domino connector runs.
- The `Notes.jar` file (included with Lotus Notes) must be available to the Domino connector.
- The `PATH` variable must specify the directory where `nnotes.dll` resides.
- Oracle Waveset with the patch shown in Certified Components for the Domino Connector must be installed.

4.1.3 Security Considerations for the Domino Connector

This section provides information about supported connections and privilege requirements:

- **Supported Connections**

If you are using the Java Connector Server, you can set the `connectorserver.usessl` property to `true` to configure the Java Connector Server to use SSL. For more information, see Deploying the Java Connector Server.

- **Required Administrative Privileges**

  To connect to the Domino resource using the Domino connector, you must specify the Administrator Account name and Admin Password configuration parameters. The Administrator password is also required to provision a user.

4.1.4 Certified Components for the Domino Connector

The Domino connector for Oracle Waveset is certified with the following components:

Table 4–3 Certified Components for the Domino Connector

Component	Requirement
Oracle Waveset	Oracle Waveset 8.1.1 Patch 4 or later
Identity Connector Framework (ICF)	ICF 1.0 or later
Target Systems	IBM Lotus Domino 8.0 and 8.5
Operating Systems	■ Microsoft Windows Server 2008 ■ Microsoft Windows Server 2003

4.1.5 Supported Languages for the Domino Connector

The Domino connector is localized in the following languages:

- Arabic
- Chinese (Simplified)
- Chinese (Traditional)
- Danish
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazilian)
- Spanish

4.2 Migrating to the Domino Connector

- Upgrading an Earlier Domino Connector
- Migrating a Domino Resource Adapter

4.2.1 Upgrading an Earlier Domino Connector

Currently, there is no automatic upgrade from an older version of the Domino connector to a newer version. You can edit the Resource and change the Domino connector version, but this option works only if there are no changes in the connector

configuration properties (shown as Resource Parameters in the Oracle Waveset Resource wizard).

In the current release, some Domino configuration properties were removed, and others were added in the Domino connector version 2.0.1. These properties must be updated manually.

The following connector configuration properties were removed: Host and Port

The following connector configuration properties were added:

- Immediate Delete - Set to True if you want the users to be deleted immediately, otherwise, Administration Process will be used to perform the deletion.
- Create Mail File In Background - Set to True if you want the User's mail file to be created in the background.
- Mail Owner Access - Set to one of the following values: 0 = Manager, 1 = DESIGNER, 2 = EDITOR
- Mail Template Name - The name of the template for the design of the mail file. If this property remains an empty string, a standard template is used.
- Disable Deny Group - Deny Group name to which is user set when disabled.
- Delete Deny Group - The name of an existing group of type `'Deny List Only'` to which the name of the deleted user is added. An empty string means do not add the user name to any group.
- Format UID - Set to True if the `UID` will be formatted like `<GUID=$UniversalId>`; otherwise, only `UniversalId` value is used as the `UID`. Set to `'true'` for Oracle Waveset.
- Sync Internet Password - Set to True to synchronize user's Internet password. The Internet password is in the user's Person document in the Domino directory. If the user changes the password for the Notes client ID, the Internet password automatically (but not immediately) changes to match it.
- Use ID Vault - If set to True, the ID Vault's password reset feature is used when changing the user's password.

You must update the connector Resource configuration XML manually. There are several methods to update the XML, such as using the Oracle Waveset debug interface or using the export/import feature. This section describes how to update the Resource XML.

To upgrade the Domino connector by updating the Resource XML, follow these steps:

1. Search for the string `bundleVersion`, which should be part of a tag such as:

```
<ConnectorRef bundleName='org.identityconnectors.domino'
bundleVersion='1.0.11'
connectorName='org.identityconnectors.domino.DominoConnector'>
```

2. Replace the version value (`'1.0.11'` in this case) with the connector version to which you are upgrading.
3. Find a tag `ResourceAttribute` with the element `name` equal to `'hostNameOrIpAddr'`. For example:

```
<ResourceAttribute name='hostNameOrIpAddr' displayName='CONN__
{ZDEwMTE=}:hostNameOrIpAddr_DISPLAY' description='CONN__
{ZDEwMTE=}:hostNameOrIpAddr_HELP' facets='provision'
value='host.example.com' required='true'
nativeName='hostNameOrIpAddr'></ResourceAttribute>
```

4. Delete the whole tag from the previous step.
5. Find a tag `ResourceAttribute` with the element `name` equal to `'hostPortNumber'`. For example:

```
<ResourceAttribute name='hostPortNumber' displayName='CONN__
{ZDEwMTE=}:hostPortNumber_DISPLAY' type='int' description='CONN__
{ZDEwMTE=}:hostPortNumber_HELP' facets='provision' value='63148'
required='true' nativeName='hostPortNumber'></ResourceAttribute>
```

6. Delete the whole tag from the previous step.
7. Add the following tags as a nested tag of the `ResourceAttributes` tag:

```
<ResourceAttribute name='immediateDelete'
    displayName='${CON__KEY}:immediateDelete_DISPLAY'
    type='boolean'
    description='${CON__KEY}:immediateDelete_HELP' facets='provision'
    value='false' nativeName='immediateDelete'>
  </ResourceAttribute>
  <ResourceAttribute name='createMailDBInBackground'
    displayName='${CON__KEY}:createMailDBInBackground_DISPLAY' type='boolean'
    description='${CON__KEY}:createMailDBInBackground_HELP' facets='provision'
    value='false' nativeName='createMailDBInBackground'>
  </ResourceAttribute>
  <ResourceAttribute name='mailOwnerAccess'
    displayName='${CON__KEY}:mailOwnerAccess_DISPLAY' type='int'
    description='${CON__KEY}:mailOwnerAccess_HELP' facets='provision'
    nativeName='mailOwnerAccess'>
  </ResourceAttribute>
  <ResourceAttribute name='mailTemplateName'
    displayName='${CON__KEY}:mailTemplateName_DISPLAY'
    description='${CON__KEY}:mailTemplateName_HELP' facets='provision'
    nativeName='mailTemplateName'>
  </ResourceAttribute>
  <ResourceAttribute name='disableDenyGroup'
    displayName='${CON__KEY}:disableDenyGroup_DISPLAY'
    description='${CON__KEY}:disableDenyGroup_HELP' facets='provision'
    nativeName='disableDenyGroup'>
  </ResourceAttribute>
  <ResourceAttribute name='deleteDenyGroup'
    displayName='${CON__KEY}:deleteDenyGroup_DISPLAY'
    description='${CON__KEY}:deleteDenyGroup_HELP' facets='provision'
    nativeName='deleteDenyGroup'>
  </ResourceAttribute>
  <ResourceAttribute name='formatUid'
    displayName='${CON__KEY}:formatUid_DISPLAY' type='boolean'
    description='${CON__KEY}:formatUid_HELP' facets='provision' value='true'
    nativeName='formatUid'>
  </ResourceAttribute>
  <ResourceAttribute name='syncInetPswd'
    displayName='${CON__KEY}:syncInetPswd_DISPLAY' type='boolean'
    description='${CON__KEY}:syncInetPswd_HELP' facets='provision'
    value='false'
    nativeName='syncInetPswd'>
  </ResourceAttribute>
  <ResourceAttribute name='useIDVault'
    displayName='${CON__KEY}:useIDVault_DISPLAY' type='boolean'
    description='${CON__KEY}:useIDVault_HELP' facets='provision' value='false'
    nativeName='useIDVault'>
  </ResourceAttribute>
```

8. Replace the `${CON__KEY}` with the connector key of the actual resource. You can find this value in the other `ResourceAttribute` tags. For example: `CONN__{ZDEwMTE=}`
9. Manually import the following files into Oracle Waveset:
   - `sample/connectors/domino-idmglue/integration.xml`
   - `sample/connectors/domino-idmglue/migration.xml`
   - `sample/connectors/domino-idmglue/postProcess.xml`
10. After updating the resource, restart Oracle Waveset to reload the resource bundle from the connector (or wait until the resource bundle cache is invalidated).

4.2.2 Migrating a Domino Resource Adapter

To migrate a Domino resource adapter to the Domino connector for Oracle Waveset, follow these steps:

1. Make sure that Oracle Waveset with the patch shown in Certified Components for the Domino Connector is installed.
2. Log in to the Oracle Waveset Administrator interface.
3. Select the Resources tab and then the Migrate Adapters tab.
4. Follow the Migration Wizard and provide values for the following new configuration options:
   - **Administration Server**: Name of the host where the Administration Server is running.
   - **Admin Account Name**: Administrator account name. For example: `Administrator/ACM`
   - **Notes Installation Folder**: Location where the Lotus Notes client is installed (where `nnotes.dll` resides). For example: `C:\Lotus\Notes\`
   - **Notes ini File**: Full path to the Lotus Notes initialization file, including the file name.

Note: During the migration from the Domino resource adapter to the Domino connector, the following options are removed: `Host` - gateway host, `TCP Port` - gateway port, `connectionLimit`, `maxThreads`, `blockCount`, `Object Class`, `updateAddrBook`, `setInternetPass`, `createDesktopClient`, `addShortName`, `User Provides Password On Change`, `removeDenyGroupsDuringDelete`, and `adminDatabase`.

4.3 Deploying the Domino Connector

Deploying the Domino connector for Oracle Waveset involves these tasks:

- Deploying the Java Connector Server
- Installing the Domino Connector
- Creating a Domino Connector Resource

4.3.1 Deploying the Java Connector Server

In a production environment, it is recommended that you deploy the Domino connector for Oracle Waveset into the Java Connector Server.

Before You Begin. The Java Connector Server requires JDK or JRE 1.5 or later to run. Make sure that the JDK or JRE is installed on the machine where you are installing the Java Connector Server and that your `JAVA_HOME` or `JRE_HOME` environment variable points to this installation.

This section describes:

- Installing and Configuring the Java Connector Server
- Running the Java Connector Server on Windows Systems
- Running the Java Connector Server on Solaris and Linux Systems

4.3.1.1 Installing and Configuring the Java Connector Server

To install and configure the Java Connector Server, follow these steps:

1. Create a new directory on the machine where you want to install the Java Connector Server. In this section, `CONNECTOR_SERVER_HOME` represents this directory.
2. Unzip the Java Connector Server package in your new directory from Step 1. The Java Connector Server package is available with the Identity Connector Framework (ICF).
3. In the `ConnectorServer.properties` file, set the following properties, as required by your deployment. The `ConnectorServer.properties` file is located in the `conf` directory.

Property	Description
`connectorserver.port`	Port on which the Java Connector Server listens for requests. The default is 8759.
`connectorserver.bundleDir`	Directory where the connector bundles are deployed. The default is `bundles`.
`connectorserver.libDir`	Directory in which to place dependent libraries. The default is `lib`.
`connectorserver.usessl`	If set to `true`, the Java Connector Server uses SSL for secure communication. The default is `false`. If you specify `true`, use the following options on the command line when you start the Java Connector Server: ■ `-Djavax.net.ssl.keyStore` ■ `-Djavax.net.ssl.keyStoreType` (optional) ■ `-Djavax.net.ssl.keyStorePassword`
`connectorserver.ifaddress`	Bind address. To set this property, uncomment it in the file (if necessary). The bind address can be useful if there are more NICs installed on the machine.
`connectorserver.key`	Java Connector Server key.

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

For more information, see Running the Java Connector Server on Windows Systems or Running the Java Connector Server on Solaris and Linux Systems.

- For all other properties, edit the `ConnectorServer.properties` file manually.

5. The `conf` directory also contains the `logging.properties` file, which you can edit if required by your deployment.

4.3.1.2 Running the Java Connector Server on Windows Systems

To run the Java Connector Server on Windows systems, use the `ConnectorServer.bat` script as follows:

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

   The `ConnectorServer.bat` script supports the following options:

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

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

4.3.1.3 Running the Java Connector Server on Solaris and Linux Systems

To run the Java Connector Server on Solaris and Linux systems, use the `connectorserver.sh` script, as follows:

1. Make sure that you have set the properties required by your deployment in the `ConnectorServer.properties` file, as described in Installing and Configuring the Java Connector Server.
2. Change to the `CONNECTOR_SERVER_HOME/bin` directory.

3. Use the `chmod` command to set the permissions to make the `connectorserver.sh` script executable.
4. Run the `connectorserver.sh` script. The script supports the following options:

Option	Description
`/run [ -Jjava-option ]`	Runs the Java Connector Server in the console. Optionally, you can specify one or more Java options. For example, to run the Java Connector Server with SSL: `./connectorserver.sh /run -J-Djavax.net.ssl.keyStore=mykeystore.jks -J-Djavax.net.ssl.keyStorePassword=password`
`/start [ -Jjava-option ]`	Runs the Java Connector Server in the background. Optionally, you can specify one or more Java options.
`/stop`	Stops the Java Connector Server, waiting up to 5 seconds for the process to end.
`/stop n`	Stops the Java Connector Server, waiting up to *n* seconds for the process to end.
`/stop -force`	Stops the Java Connector Server. Waits up to 5 seconds and then uses the `kill -KILL` command, if the process is still running.
`/stop n -force`	Stops the Java Connector Server. Waits up to *n* seconds and then uses the `kill -KILL` command, if the process is still running.
`/setKey key`	Sets the Java Connector Server key. The `connectorserver.sh` script stores the hashed value of *key* in the `connectorserver.key` property in the `ConnectorServer.properties` file.

4.3.2 Installing the Domino Connector

To deploy the Domino connector for Oracle Waveset into the Java Connector Server, follow these steps:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the Domino Connector.
2. Stop the Java Connector Server.
3. Copy the Domino connector bundle into the Java Connector Server `CONNECTOR_SERVER_HOME\bundles` directory.
4. Copy the `Notes.jar` file from the Lotus Notes installation directory to the `CONNECTOR_SERVER_HOME\lib` directory.
5. Ensure that the `PATH` variable specifies the directory where `nnotes.dll` resides.
6. Start the Java Connector Server.

4.3.3 Creating a Domino Connector Resource

To create a Domino connector resource, follow these steps:

1. Log in to the Oracle Waveset Administrator interface.
2. Add the Java Connector Server to Oracle Waveset by selecting Configure, Connector Servers, and then New.

3. Create the Domino connector resource by following the Create Domino Connector Resource wizard.

   Note. To provision a user, you must first provide the Administrator password.

4. Specify values for the configuration parameters as described in Domino Connector Configuration Parameters.

For additional information about creating resources, see "Understanding and Managing Waveset Resources" in the *Oracle Waveset 8.1.1 Business Administrator's Guide* in the following library:

`http://docs.oracle.com/cd/E19225-01/index.html`

4.4 Using the Domino Connector

- Object Classes and Attributes Supported by the Domino Connector
- Domino Connector Sample Forms
- Executing Before and After Actions

4.4.1 Object Classes and Attributes Supported by the Domino Connector

This section provides the following information about the object classes and attributes supported by the Domino connector for Oracle Waveset:

- ACCOUNT Object Class
- GROUP Object Class
- Attribute Mapping Changes

4.4.1.1 ACCOUNT Object Class

The Domino connector for Oracle Waveset supports `__ACCOUNT__`, People (denotes the same object class), and the attributes shown in the following table.

Table 4–4 ACCOUNT Object Class Attributes

Resource User Attribute	Type	Required	Description
`__CURRENT_PASSWORD__`	GuardedString	No	Current password. Not creatable, not readable, and not returned by default.
`__ENABLE__`	Boolean	No	Enables (`true`) or disables (`false`) a user by removing or adding the user to `DenyGroups` (if provided) or by setting the `CheckPassword` attribute to `Lockout`.
`__NAME__`	String	Yes	Name
`__PASSWORD__`	GuardedString	No	Password. Not readable and not returned by default.
`AltOrgUnit`	String	No	Organizational unit for the user in the alternate language. Can be multi-valued.
`AltFullName`	String	No	User's full name in the user's native language
`AltFullNameLanguage`	String	No	Language associated with the alternate full name.
`Assistant`	String	No	Name of an assistant.

Table 4–4 (Cont.) ACCOUNT Object Class Attributes

Resource User Attribute	Type	Required	Description
CalendarDomain	String	No	Domain name for the calendar.
CellPhoneNumber	String	No	User's cell phone number.
certifierIDFile	String	No	Path to the certifier ID file. Not readable and not returned by default.
CertifierOrgHierarchy	String	No	Path of certifier's organization hierarchy, such as /US1 (overrides value on resource).
CheckPassword	Integer	No	Indicates whether to check the user's password: 0 = No check. 1 = Check. 2 = Disable user.
Children	String	No	Name or names of the employee's children.
City	String	No	City of the user's home address.
Comment	String	No	Comment about the user.
CompanyName	String	No	User's company name.
Country	String	No	Country of the user's home address.
credentials	GuardedString	No	Password for the certifier ID file (overrides value on resource). Not readable and not returned by default.
defaultPasswordExp	Integer	No	Number of days for new certificates to be issued (create, recertify operations). Not readable and not returned by default.
DenyGroups	String	No	List of users that are to be denied access to the resource. Not returned by default
Department	String	No	The department name or number of the user.
DisplayName	String	No	User's displayed name.
EmployeeID	String	No	Unique employee ID for the user.
FirstName	String	Yes	User's first name.
FullName	String	No	User's full name. Not creatable and not updatable. Can be multi-valued.
GroupList	String	No	List of groups. Can be multi-valued.
HomeFAXPhoneNumber	String	No	User's home FAX phone number
HTTPPassword	GuardedString	No	Password to be used when accessing a Notes server from a web browser or other HTTP client. Not readable and not returned by default.
idFile	String	No	Fully qualified path to the ID file. Required for a create operation but not used afterwards. Not updatable, not readable, and not returned by default.
InternetAddress	String	No	User's internet address.

Table 4–4 (Cont.) ACCOUNT Object Class Attributes

Resource User Attribute	Type	Required	Description
JobTitle	String	No	User's job title.
LastModified	Long	No	Last date and time the user was modified. Not creatable and not updatable.
LastName	String	Yes	User's last name.
Location	String	No	Office location or mail stop
MailAddress	String	No	User's e-mail address.
MailDomain	String	No	Domain name of user's mail server
MailFile	String	No	Name of the mail file. For example: MAIL\JSMITH
MailQuotaSizeLimit	Integer	No	Specifies the maximum size of the user's mail database. If you specify a value less than 1000, then the maximum size is in megabytes (MB). If the value is 1000 or greater, then the maximum size is expressed in bytes. Values between 1001 and 1023 are rounded up to 1024 bytes. The proxy administrator must be listed as an Administrator in the Server document to set this attribute.
MailQuotaWarningThreshold	Integer	No	Specifies the size of a user's mail database at which point a warning about the size of the database is generated. If you specify a value less than 1000, then the threshold is in megabytes (MB). If the value is 1000 or greater, then the threshold is expressed in bytes. Values between 1001 and 1023 are rounded up to 1024 bytes. The proxy administrator must be listed as an Administrator in the Server document to set this attribute.
MailServer	String	No	User's mail server name.
MailTemplateName	String	No	Name of mail template. Valid only during create.
Manager	String	No	User's manager.
MiddleInitial	String	No	User's middle initial with a trailing period.
NetUserName	String	No	User's network account name.
objectGUID	String	No	User's Notes ID. Not creatable and not updatable.
OfficeCity	String	No	City of the user's work address.
OfficeCountry	String	No	Country of the user's work address.
OfficeFAXPhoneNumber	String	No	FAX number of the user's work address.
OfficeNumber	String	No	Office number of the user's work address.
OfficePhoneNumber	String	No	Phone number of the user's work address.

Table 4–4 (Cont.) ACCOUNT Object Class Attributes

Resource User Attribute	Type	Required	Description
OfficeState	String	No	State or province of the user's work address.
OfficeStreetAddress	String	No	Street address of the user's work address.
OfficeZIP	String	No	Postal code of the user's work address.
orgUnit	String	No	User's organizational unit.
PasswordChangeInterval	Integer	No	Number of days after which the user must supply a new password.
PasswordGracePeriod	Integer	No	Number of days after the password has expired before the user is locked out.
PhoneNumber	String	No	User's home telephone number.
PhoneNumber_6		No	User's home telephone number in six digits.
Policy	String	No	Name of the Domino explicit policy to be assigned to the user. When set, this policy can modify or override other user attribute values. Applies only to Domino 7.0 and later.
Profiles	String	No	Profile assigned to the user. This value overrides any profile specified as a resource parameter. Applies only to Domino 7.0 and later.
Recertify	Boolean	No	Indicates whether to recertify a user. Not readable and not returnable by default.
RoamCleanPer	Integer	No	If RoamCleanSetting is 1, specifies the number of days between cleaning.
RoamCleanSetting	Integer	No	Specifies when Domino cleans up the user's roaming files. Values can be: 0 = Never. 1 = Periodically. 2 = When the Domino server shuts down. 3 = Prompt the user.
RoamingUser	String	No	When set to 1, specifies that the user is a roaming user.
RoamRplSrvrs	String	No	List of servers where the user's roaming files are to be replicated.
RoamSrvr	String	No	Server where the user's roaming files are to be located.
RoamSubdir	String	No	Directory that will contain the user's roaming files.
SametimeServer	String	No	Hierarchical name of the user's same time server.
ShortName	String	No	Short user name commonly used by a foreign mail system.
Spouse	String	No	Name of the user's spouse.

Table 4–4 (Cont.) ACCOUNT Object Class Attributes

Resource User Attribute	Type	Required	Description
State		No	State or province in the user's home address.
StreetAddress	String	No	Address of the user's home address.
Suffix	String	No	User's generational qualifier
Title	String	No	User's title
WebSite	String	No	User's web site.
x400Address	String	No	User's x400 address.
Zip	String	No	Postal code of the user's home address.

4.4.1.2 GROUP Object Class

The Domino connector for Oracle Waveset supports __GROUP__, Groups, Group (denotes the same object class) and the attributes shown in the following table.

Table 4–5 GROUP Object Class Attributes

Name	Type	Required	Not Creatable	Not Updatable
__NAME__	String	Yes	-	-
Comments	String	No	-	-
DisplayName	String	No	Yes	Yes
GroupName	String	No	-	-
GroupTitle	String	No	-	-
GroupType	String	No	-	-
LastModified	Long	No	Yes	Yes
ListCategory	String	No	-	-
ListDescription	String	No	-	-
ListName	String	No	-	-
Members	String	No	-	-
objectGUID	String	No	Yes	Yes

4.4.1.3 Attribute Mapping Changes

Post processing (postProcess.xml) changes the attribute mapping shown in the following table.

Table 4–6 Attribute Mapping Changes

Oracle Waveset Attribute Name	Oracle Waveset Attribute Type	Domino Connector Attribute Name	Domino Connector Attribute Type
NotesGroups	String	GroupList	String
orgUnit	String	OrgUnit	String
AlternateOrgUnit	String	AltOrgUnit	String

Table 4–6 (Cont.) Attribute Mapping Changes

Oracle Waveset Attribute Name	Oracle Waveset Attribute Type	Domino Connector Attribute Name	Domino Connector Attribute Type
MailTemplate	String	MailTemplateName	String
dbQuotaSizeLimit	String	MailQuotaSizeLimit	String
dbQuotaWarningThreshold	String	MailQuotaWarningThreshold	String
lastModified	String	LastModified	Long
RoamCleanPer	String	RoamCleanPer	Integer

4.4.2 Domino Connector Sample Forms

The Domino connector for Oracle Waveset includes these sample forms:

- DominoActiveSyncForm.xml
- Dominogroupcreate.xml
- Dominogroupupdate.xml

For information about customizing a form to meet your specific requirements, see "Customizing Forms" in the *Oracle Waveset 8.1.1 Deployment Reference* in the following library:

http://docs.oracle.com/cd/E19225-01/index.html

4.4.3 Executing Before and After Actions

The Domino connector includes the new DominoConnector-Integration configuration object to allow you to control how before and after actions are executed. For the layout of this configuration object, see Example 4–1.

To edit the DominoConnector-Integration configuration object, access the Oracle Waveset Debug Page, as described in "Editing Waveset Configuration Objects" in the *Oracle Waveset 8.1.1 Business Administrator's Guide* in the following library:

http://docs.oracle.com/cd/E19225-01/index.html

The Domino connector supports the Lotus Script and cmd shell types of actions. If an action does not have an associated script language specified, the Domino connector uses cmd as the default action:

```
<Attribute name='defaultActionType' value='cmd'/>
```

The cmd default value ensures the same behavior as the superseded Domino resource adapter. However, if required by your deployment, you can edit the DominoConnector-Integration configuration object to change this default value.

If you do not specify whether an action should run on a resource or connector, the Domino connector uses the following mapping:

```
<Attribute name='actionExecModes'>
  <Map>
    <MapEntry key='lotusscript' value='resource'/>
    <MapEntry key='cmd' value='connector'/>
  </Map>
</Attribute>
```

Caution: Do not change this mapping; otherwise, before and after actions will not work properly.

Example 4–1 DominoConnector-Integration Configuration Object

```
<Waveset>
  <Configuration name="DominoConnector-Integration">
    <Extension>
      <List>
        <Object>
          <Attribute name='version'>
            <List>
              <Object>
                <Attribute name='versionSegment' value='major' />
                <Attribute name='operator' value='eq' />
                <Attribute name='operand' value='1' />
              </Object>
              <Object>
                <Attribute name='versionSegment' value='minor' />
                <Attribute name='operator' value='eq' />
                <Attribute name='operand' value='0' />
              </Object>
            </List>
          </Attribute>
          <Attribute name='content'>
            <Object>
              <Attribute name='actionExecModes'>
                <Map>
                  <MapEntry key='lotusscript' value='resource'/>
                  <MapEntry key='cmd' value='connector'/>
                </Map>
              </Attribute>
              <Attribute name='defaultActionType' value='cmd'/>
            </Object>
          </Attribute>
        </Object>
      </List>
    </Extension>
  </Configuration>
</Waveset>
```

4.5 Troubleshooting the Domino Connector

Use the Oracle Waveset debug pages to set trace options on the following class:

`org.identityconnectors.framework.impl.api.LoggingProxy`

To set the logging options for the Java Connector Server, edit the properties in the `CONNECTOR_SERVER_HOME\conf\logging.properties` file.

4.6 Known Issues for the Domino Connector

- Bug 12640400: Migration From Domino Adapter Requires Manual Steps
- Bug 12531662: Changing Password Using ID Vault is Not Supported

4.6.1 Bug 12640400: Migration From Domino Adapter Requires Manual Steps

If you migrate a Domino adapter in Oracle Waveset to the Domino connector, some account attribute mappings are not consistent. After the migration is finished, the "password" attribute is left from the original Domino Adapter.

Workaround. You must remove the "password" attribute mapping manually, as follows:

1. In Oracle Waveset, go to the "Account Attributes "mapping configuration of the migrated resource and find the "password" attribute.
2. Select the "password" attribute by checking its checkbox.
3. Click the "Remove Selected Attribute(s)" button.

4.6.2 Bug 12531662: Changing Password Using ID Vault is Not Supported

If the `Use ID Vault` configuration parameter is set to true and you try to change a password, the Domino connector throws an exception with a message stating that the operation is not supported.

Workaround. None.

5

Oracle Waveset Connector for Microsoft Exchange

This chapter includes the following information about the Exchange connector for Oracle Waveset:

5.1 About the Exchange Connector

5.1.1 Overview of the Exchange Connector

The Exchange connector is a .NET connector that supports provisioning and reconciliation for target systems running Microsoft Exchange Server On-Premise 2007 and On-Premise 2010.

The Exchange connector also extends the functionality of the Active Directory connector. The Exchange connector bundle includes both the Exchange connector and Active Directory connector libraries (DLL files). If you use the Exchange connector to provision Active Directory, the operations and deployment considerations for the Active Directory connector also apply to the Exchange connector. For more information, see Chapter 2, "Oracle Waveset Connector for Microsoft Active Directory".

Other considerations for the Exchange connector are:

- The Exchange connector uses Windows PowerShell to manage the Exchange Server target systems. Therefore, the appropriate PowerShell `cmdlet` set must be available to the connector.

- The Exchange connector operates in the context of the .NET Connector Framework, which in turn requires an application to execute. Therefore, by default, Oracle provides (and recommends) the .NET connector server to run the Exchange connector and Active Directory connector. For more information, see Deploying the Exchange Connector.
- The Exchange connector supports agentless target deployment; that is, an agent is not required.
- The Exchange connector supersedes the Exchange resource adapter and earlier versions of the Exchange connector. For migration information, see Migrating to the Exchange Connector.

Additional information in this section includes:

- Exchange Connector Features
- Exchange Connector Resource Configuration Parameters

5.1.1.1 Exchange Connector Features

The Exchange connector supports the operations shown in the following table. The Exchange connector also supports the operations for Active Directory, as described in Active Directory Connector Features.

Table 5–1 Exchange Connector Operations

Operation	Description
Account provisioning	Operations include create, read, update, and delete objects.
Reconciliation	Both full and incremental reconciliation are supported. Reconciliation can perform the following functions: ■ Detect new mailboxes or new mail users ■ Detect changes in mailbox attribute values or mail user attribute values ■ Correlate mailboxes or mail users with Oracle Waveset users ■ Detect mailboxes or mail users that are not associated with Oracle Waveset users Active Sync polls for changes to a resource and detects incremental changes in real time. Active Sync is performed by the Active Directory connector. For additional information about Active Sync, see the *Oracle Waveset 8.1.1 Deployment Guide* in the following library: `http://docs.oracle.com/cd/E19225-01/index.html`
Before and after actions	Before and after actions use scripts to perform activities during a user create, update, or delete request. Script execution is performed by the Active Directory connector. For more information, see Configuring Before and After Actions for the Active Directory Connector.

5.1.1.1.1 Provisioning and Reconciliation Across Multiple Domains

The Exchange connector supports reconciliation and provisioning of mailboxes for users across multiple Microsoft Active Directory domains. For example, users on ChildDomain1, ChildDomain2, ParentDomain, and PeerDomain can have mailboxes in the same Exchange Server installed on the parent domain. Oracle Waveset can reconcile and provision mailboxes for the users who belong to each of these domains, with a resource pointing to the parent domain where Exchange Server is installed.

Note: Latency in Replication of Data Between Domain Controllers and the Global Catalog Server

The Exchange connector gets user and distribution group information from the domain controllers. Information residing in different domain controllers must be replicated among each other. At certain times, there can be a delay in this replication, and complete user or distribution group information might not be pulled into Oracle Waveset by the connector. In these situations, it is recommended to search for the entity information again.

By default, the user forms cache Active Directory and Exchange groups. After you create a new resource, clear the cache. To clear the cache, navigate to debug page and click Clear Resource Object List Cache. After the cache is cleared, Oracle Waveset obtains information about the groups from the target system. Subsequently, Oracle Waveset obtains this information from cache.

5.1.1.2 Exchange Connector Resource Configuration Parameters

When you configure a resource for the Exchange connector on a Windows target system, the connector uses the Active Directory connector configuration parameters, described in Active Directory Connector Resource Configuration Parameters.

The Exchange connector also requires the additional configuration parameters described in the following table.

Table 5–2 Exchange Connector Resource Configuration Parameters

Parameter Name	Type	Required	Description
AuthenticationMechanism	String	Yes	This entry is used when the connector is configured against Exchange 2010 to remotely connect to the Exchange Server. Default value: `Kerberos` Do not modify this entry.
Exchange Server Type	String	Yes	Version of the Exchange Server target. The choices are: ▪ OnPremise2007 ▪ OnPremise2010
ExchangeServerHost	String	No	Hostname of the computer hosting Exchange Server 2010. This is required only if ExchangeServerType is set to OnPremise2010.
ExchangeUser	String	No	User name of the service account having minimum privileges. Format: DomainName\UserName This is required only if ExchangeServerType is set to OnPremise2010.
ExchangeUserPassword	String	No	Valid password for user specified for the ExchangeUser parameter. This is required only if ExchangeServerType is set to OnPremise2010.

Table 5–2 (Cont.) Exchange Connector Resource Configuration Parameters

Parameter Name	Type	Required	Description
UseSSLForRemotePowerShell	String	Yes	This entry is used when the connector is configured against Exchange 2010 to remotely connect to the connector. Default value: `false` Do not modify this entry.

5.1.2 Security Considerations for the Exchange Connector

- Secure Communication to the Target System
- Administrator Account Considerations for Exchange Server

For Active Directory, see Security Considerations for the Active Directory Connector.

5.1.2.1 Secure Communication to the Target System

On the Exchange connector side, secure communication is ensured by the API. Any bind to the directory is secured by the Windows Security Support Provider Interface (SSPI).

Communication to the Exchange Server is secured by Windows PowerShell.

The communication between the .NET Connector Framework and Oracle Waveset is encrypted by the framework, but it is also recommended that you use an SSL connection.

5.1.2.2 Administrator Account Considerations for Exchange Server

Depending on the Exchange Server version you are using, ensure the administrator account meets the following requirements:

- Privileges for Exchange 2007 Service Account
- Privileges for Exchange 2010 Service Account

5.1.2.2.1 Privileges for Exchange 2007 Service Account

The minimum privileges required for a Exchange 2007 service account to manage recipients (UserMailbox and MailUser) are:

- The service account must be a member of Exchange Recipient Administrators group.

  For more information, see `http://technet.microsoft.com/en-us/library/aa996881%28v=exchg.80%29.aspx`.

- If you want to add a recipient to a distribution group or remove a recipient from a distribution group, then the service account must also be a member of Account Operators group in the domain where the distribution group exists.

  For more information, see `http://technet.microsoft.com/en-us/library/bb124340%28v=exchg.80%29.aspx` and `http://technet.microsoft.com/en-us/library/aa997627%28v=exchg.80%29.aspx`.

5.1.2.2.2 Privileges for Exchange 2010 Service Account

The minimum privilege required for a Exchange 2010 service account to manage recipients (UserMailbox and MailUser) is:

- The service account must be a member of Recipient Management group.

  For more information, see http://technet.microsoft.com/en-us/library/dd298028%28v=exchg.141%29.aspx.

5.1.3 Certified Components for the Exchange Connector

The Exchange connector is certified with the following components:

Table 5–3 Exchange Connector Certified Components

Component	Requirement
Oracle Waveset	Oracle Waveset 8.1 Update 1 Bundle Patch 8 or later
Identity Connector Framework (ICF)	ICF 1.2 or later
Microsoft .NET Framework	Microsoft .NET Framework 3.5 **Note:** To prevent a memory leak problem with Microsoft .NET Framework 3.5, apply the hotfix described in the following article: http://support.microsoft.com/kb/981575
Target Systems	The target system can be any one or a combination of the following: ■ Microsoft Exchange 2007 SP1, SP2, SP3 (64-bit) ■ Microsoft Exchange 2010 RTM, SP1, SP2, SP3 (64-bit)

5.1.4 Supported Languages for the Exchange Connector

The Exchange connector is localized in the following languages:

- Arabic
- Chinese (Simplified)
- Chinese (Traditional)
- Danish
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazilian)
- Spanish

5.2 Migrating to the Exchange Connector

- Migrating an Earlier Exchange Connector
- Migrating an Exchange Resource Adapter

- Post Migration Task

5.2.1 Migrating an Earlier Exchange Connector

To migrate from an earlier version of the Exchange connector, follow these steps:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the Exchange Connector.
2. Log in to the Oracle Waveset Administrator interface.
3. In the Edit Resource wizard, specify the new Exchange connector version and configure the Resource Parameters, as required for your deployment.

5.2.2 Migrating an Exchange Resource Adapter

Before You Get Started: Install and configure the latest version of the connector server, as described in Installing, Configuring, and Running the Connector Server.

To migrate an Exchange resource adapter to the Exchange connector, follow these steps:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the Exchange Connector.
2. Log in to the Oracle Waveset Administrator interface.
3. Go to the Migrate Adapters page.
4. Select the Resource you want to migrate from.
5. Select the type of the connector you want to convert to.
6. On the next page, select the Exchange connector version and the connector server to use.
7. Provide the Active Directory Domain Name and Exchange Server Type.
8. Click Convert.

5.2.3 Post Migration Task

After migrating an earlier Exchange connector or Exchange resource adapter, perform the following task:

1. In the Exchange connector userForm.xml file, replace "Exchange" with "Windows Active Directory" in the following field:

```
<contains>
<ref>accountInfo.typeNames</ref>
<s>Exchange</s>
<contains>
```

2. In the postProcess.xml file, replace "Exchange" with "Windows Active Directory" in the following line:

```
<MapEntry key='typeString' value='Exchange'/>
```

3. Import the userForm.xml and postProcess.xml files into Oracle Waveset, as follows:
   a. Click the Configure tab.
   b. Click Import Exchange File.

c. Select the XML files and then click Import.

5.3 Deploying the Exchange Connector

5.3.1 Exchange Connector Deployment Architecture

The connector uses Exchange-related PowerShell cmdlets to perform recipient administration activities on the Exchange Server. The connector supports User, UserMailbox, and MailUser recipient types. Connector server is mandatory for both Exchange 2007 and Exchange 2010 target system versions.

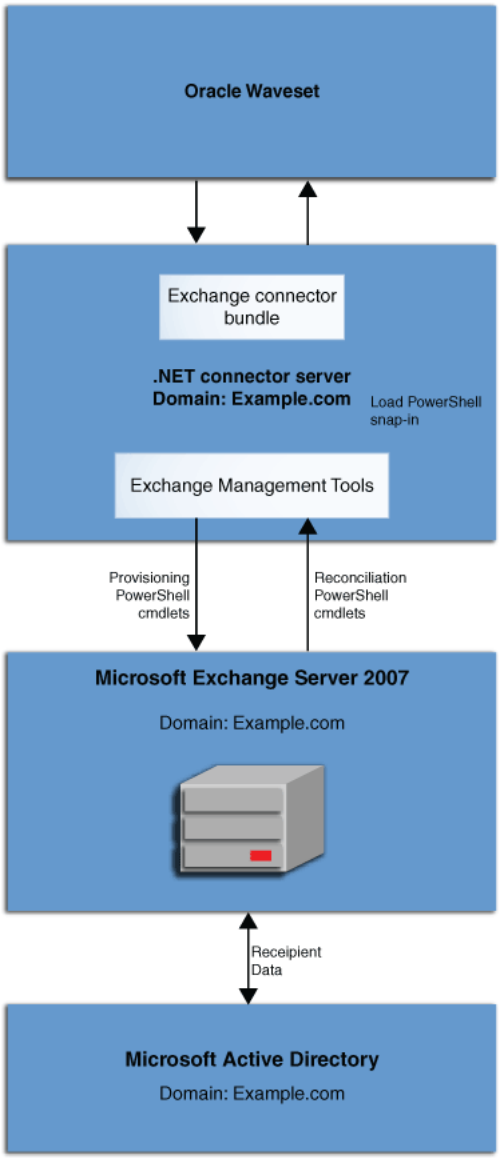
Note: You can also use the Exchange connector to manage Active Directory users. To manage Active Directory users, select RecipientType as User in the user form. In this case, the Exchange connector only creates a user in Active Directory. The Exchange connector will *not* create any UserMailbox or mark this Active Directory user as a MailUser. For more information, see Oracle Waveset Connector for Microsoft Active Directory.

The following sections discuss details specific to Exchange resource only.

For more information about recipient types, see `http://technet.microsoft.com/en-us/library/bb201680%28v=exchg.141%29.aspx`.

Figure 5–1 shows the architecture of the connector supporting Exchange Server 2007. In this architecture diagram, the connector server is installed on a different computer in the same domain as that of the Exchange Server computer. You can also install the connector server on the same computer hosting Exchange Server.

Figure 5–1 Architecture of the Connector Supporting Exchange Server 2007

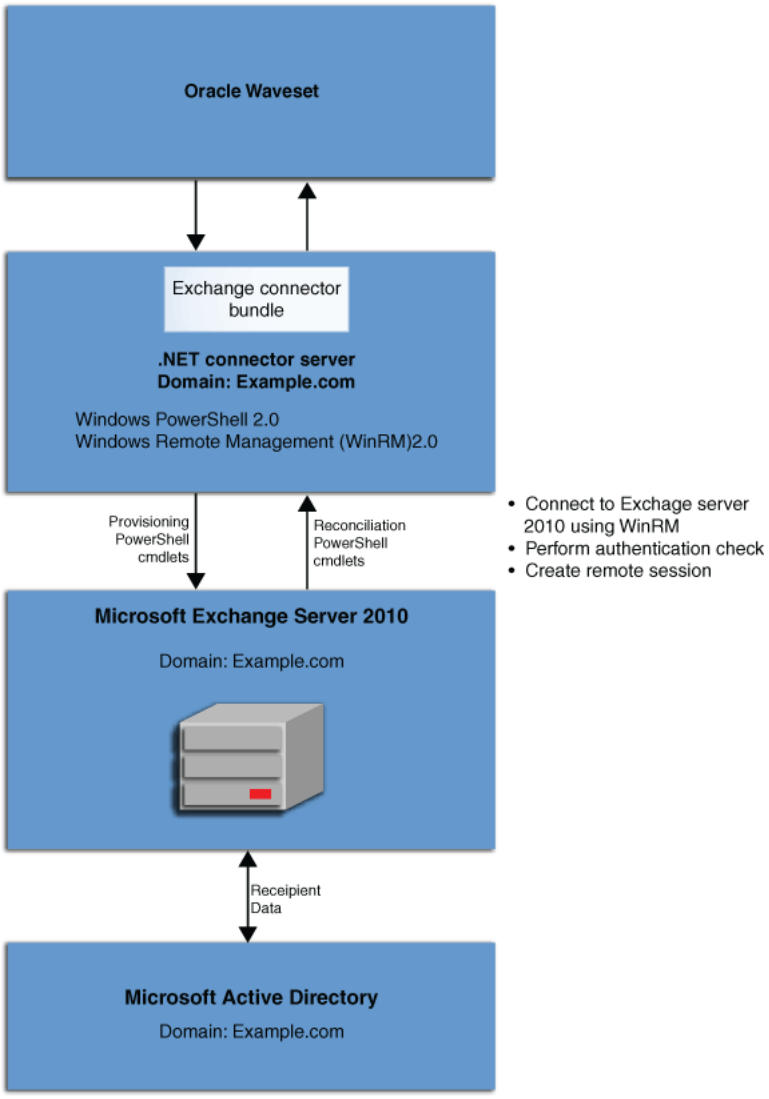


Oracle Waveset communicates with Exchange Server 2007 via connector bundle. The connector bundle is deployed on a Windows computer with the connector server installed. To communicate with Exchange Server 2007, the connector loads the Microsoft.Exchange.Management.PowerShell.Admin snap-in locally to create a runspace, which is the environment for running PowerShell cmdlets. This snap-in becomes available when Exchange Management Tools are installed. For this reason, Exchange Management Tools must be installed on the Windows computer hosting connector server.

For more information on hardware requirements, installing, and configuring connector server, see Installing, Configuring, and Running the Connector Server.

Figure 5–2 shows the architecture of the connector supporting Exchange Server 2010. In this architecture diagram, the connector server is installed on a different computer in the same domain as that of the Exchange Server computer. You can also install the connector server on the same computer hosting Exchange Server.

Figure 5–2 Architecture of the Connector Supporting Exchange Server 2010



Oracle Waveset communicates with Exchange Server 2010 via connector bundle. The connector bundle is deployed on a Windows computer with the connector server installed. To communicate with Exchange Server 2010, Oracle Waveset uses remote Shell, which in turn uses Windows PowerShell 2.0 and Windows Remote Management (WinRM) 2.0 without the need for Exchange Management Tools. Therefore, Exchange Management Tools are not required to be installed on the connector server for Exchange Server 2010. For more information, see the following topic on Remote Exchange Management at:

http://technet.microsoft.com/en-in/library/dd297932%28v=exchg.141%29.aspx

Run the **Enable-PSRemoting** cmdlet to configure the Exchange Server computer to receive Windows PowerShell remote commands that are sent by using the WS-Management technology. For more information about the Enable-PSRemoting cmdlet, see:

http://technet.microsoft.com/en-us/library/hh849694.aspx

For more information on hardware requirements, installing, and configuring connector server, see Installing, Configuring, and Running the Connector Server.

5.3.2 Downloading the Exchange Connector

The Exchange connector is available on the Oracle Identity Manager Connector Downloads page:

http://www.oracle.com/technetwork/middleware/id-mgmt/downloads/connectors-101674.html

5.3.3 Installing, Configuring, and Running the Connector Server

The **connector server** is an application that enables remote execution of the Exchange connector. As the Exchange connector is implemented in .NET, it requires a .NET connector server.

The connector server can either be installed on the same computer as that of the Exchange Server or on a different computer in the same domain as that of the Exchange Server. For more information, see Exchange Connector Deployment Architecture.

This section contains the following topics:

- Pre-requisites for the Connector Server
- Installing the Connector Server
- Configuring the Connector Server
- Enabling Logging
- Configuring Log File Rotation
- Running the Connector Server

5.3.3.1 Pre-requisites for the Connector Server

The following pre-requisites and requirements are recommended for the connector server:

- The computer hosting the connector server has Intel Dual-Core Processor, 2 GHz with 4 GB RAM or a computer with similar configuration.

  If you have a computer dedicated to the connector server, then 2 GB RAM is sufficient.

- Before you install the connector server, ensure that you have installed .NET Framework 3.5 SP1 on the same computer where you are installing the connector server.

  In addition, you must install the following patch:

  `http://support.microsoft.com/kb/981575`

  The connector server need not be installed on the Exchange server target system. It can be installed either on the Exchange server or on a system that belongs to the same domain as that of the Exchange server. However, it is recommended to deploy connector server on a separate system that belongs to the same domain as that of the Exchange server.

- If you are using Exchange Server 2007, then you must install Exchange Management Tools on the computer hosting the connector server. This is a mandatory requirement.

- If you are using Exchange Server 2010, then the pre-requisites and requirements must be met for remote Shell as mentioned in the Remote Exchange Management page at:

  http://technet.microsoft.com/en-in/library/dd297932%28v=exchg.141%29.aspx

  Windows 2008 R2 is recommended as the operating system for the computer hosting the connector server as it includes all the necessary pre-requisites and requirements.

5.3.3.2 Installing the Connector Server

To install the connector server:

1. Download the connector server package (a zip file such as Connector_Server_111200.zip) from the Oracle Identity Manager Connector Downloads page at:

   http://www.oracle.com/technetwork/middleware/id-mgmt/downloads/connectors-101674.html

2. Extract the contents of the connector server package and locate the ServiceInstall-*version*.msi file, such as ServiceInstall-1.4.0.0.msi.

3. Install the connector server by running the ServiceInstall-1.4.0.0.msi file.

   If the **Setup Type - Typical** option is used during the installation, then the connector server will be installed at the C:\Program Files\Identity Connector\Connector Server directory.

   Note: In this guide, *CONNECTOR_SERVER_HOME* represents the C:\Program Files\Identity Connector\Connector Server directory.

4. Upon successful installation, the connector server is registered as a Windows service and will be started automatically.

5. Stop the connector server Windows service.

5.3.3.3 Configuring the Connector Server

To configure the connector server:

1. Open the connectorserver.exe.config file located in the *CONNECTOR_SERVER_HOME* directory. In the connectorserver.exe.config file, set the following properties, as required by your deployment.

Property	Description
connectorserver.port	Port on which the connector server listens for requests. Default value: `8759`
connectorserver.usessl	If set to **`true`**, the connector server uses SSL for secure communication with Oracle Waveset. If this property is set to **true,** then you must set the corresponding property in Exchange Connector Server IT resource to `true`. Default value: `false`.
Certificatestorename	If the connectorserver.usessl property is set to true, then this property should point to your certificate store name.

Property	Description
connectorserver.key	Connector server key. See Step 2 for information about setting this value.

2. Set The connector server key in the connectorserver.exe.config file, as follows:

   Note: This key value must be mentioned in the Exchange Connector Server IT resource property.

   a. Open a command prompt and navigate to *CONNECTOR_SERVER_HOME* directory.

   b. Run the ConnectorServer.exe /setKey command.

      This displays the prompt **Enter Key:**

   c. Enter an appropriate key and press Enter.

      This displays the prompt **Confirm Key:**

   d. Enter the same key to confirm and press Enter.

      This displays the message Key Updated.

5.3.3.4 Enabling Logging

The Exchange connector uses the built-in logging mechanism of the .NET framework. Logging for the Exchange connector is not integrated with Oracle Waveset. The log level is set in the connector server configuration file (ConnectorServer.exe.config).

By default, logging is not enabled for the connector. To enable logging:

1. Navigate to *CONNECTOR_SERVER_HOME* directory. The default directory is C:\Program Files\Identity Connectors\Connector Server.

   The ConnectorServer.exe.config file must be present in this directory.

2. Search and locate the tag **<add name="myListener"** under the <listeners> tag.

3. The connector logs all information in the file indicated by the **initializeData** parameter. The default value is `c:\connectorserver.log.`

   Edit this value as per your deployment needs. As the connector server runs using the service account, ensure the service account has write permissions on the log location and on the log file. Otherwise, there would be no logs generated even if you enable logging.

4. In the ConnectorServer.exe.config file, add the lines shown in bold text:

```
<system.diagnostics>
  <trace autoflush="true" indentsize="4">
    <listeners>
      <remove name="Default" />
      <add name="myListener" type="System.Diagnostics.TextWriterTraceListener"
initializeData="c:\connectorserver.log" traceOutputOptions="DateTime">
        <filter type="System.Diagnostics.EventTypeFilter"
initializeData="Information" />
      </add>
    </listeners>
  </trace>
  <switches>
```

```
    <add name="ExchangeSwitch" value="4" />
  </switches>
</system.diagnostics>
```

The `value="4"` sets the log level to Verbose. This value can be set as follows:

Table 5–4 Log Levels

Value	Log Level
value="4" or value="Verbose"	Verbose level. Most granular.
value="3" or value="Information"	Information level.
value="2" or value="Warning"	Warning level.
value="1" or value="Error"	Error level.
value="0"	No logging.

5.3.3.5 Configuring Log File Rotation

Information about events that occur during the course of reconciliation and provisioning operations are stored in a log file. As you use the connector over a period time, the amount of information written to a log file increases. If no rotation is performed, then log files become huge.

To avoid such a scenario, perform the procedure described in this section to configure rotation of the log file.

To configure rotation of a log file on a daily basis:

1. Log in to the computer that is hosting the connector server.
2. Stop the connector server.
3. Back up the ConnectorServer.exe.config file. The default location of this file is C:\Program Files\Identity Connectors\Connector Server.
4. In a text editor, open the ConnectorServer.exe.config file for editing.
5. Search for the <listeners> and </listeners> elements and replace the text between these elements with the following:

```
<remove name="Default" />
<add name="FileLog"
type="Microsoft.VisualBasic.Logging.FileLogTraceListener,Microsoft.VisualBasic,
Version=8.0.0.0,Culture=neutral,PublicKeyToken=b03f5f7f11d50a3a"
initializeData="FileLogWriter"
traceOutputOptions="DateTime"
BaseFileName="ConnectorServerDaily"
Location="Custom"
CustomLocation="C:\ConnectorServerLog\"
LogFileCreationSchedule="Daily">
<filter type="System.Diagnostics.EventTypeFilter"
initializeData="Information"/>
</add>
```

6. Save the file and close it.
7. Start the connector server.

See Also: The following URL for more information about configuring log file rotation:

http://msdn.microsoft.com/en-us/library/microsoft.visualbasic.logging.filelogtracelistener.aspx

5.3.3.6 Running the Connector Server

To run the connector server, perform one of the following steps depending on the Exchange Server version:

- If you are using Exchange Server 2007:
  a. Login to computer hosting the connector server.

     The login user must have permissions to perform the following steps.

  b. Open Windows services explorer. To do so:

     Click the **Start** button, then click to **Run...** Enter `Services.msc` and click **OK.**

  c. Locate the **Connector Server** service.

  d. Right-click on the service and click **Properties.**

  e. Click the **Log On** tab and select **This Account.**

  f. Click **Browse** to choose the service account having minimum privileges as described in Privileges for Exchange 2007 Service Account Then, enter password for this service account.

  g. Click **OK.**

  h. With this service selected, click **Run.**

  After the above steps are completed successfully, the connector server runs with the service account that has the minimum privileges to perform recipient management tasks on Exchange Server 2007.

  Note: The above steps are mandatory and must be completed successfully. This is because the Exchange connector uses the credentials of the user who starts the connector server to communicate with Exchange Server 2007. The username and password information provided in the Exchange resource is not used.

- If you are using Exchange Server 2010:
  a. Login to computer hosting the connector server.

     The login user must have permissions to perform the following steps.

  b. Open Windows services explorer. To do so:

     Click the **Start** button, then click to **Run...** Then, enter `Services.msc` and click **OK.**

  c. Locate the **Connector Server** service and click **Run.**

Note: The Exchange connector uses the user credentials provided in the Exchange resource to communicate with Exchange Server. Therefore, any user can start the connector server.

Alternatively, the connector server can be started by the service account having minimum privileges as described in Privileges for Exchange 2010 Service Account.

5.3.4 Installing the Exchange Connector

To install the Exchange connector bundle in the connector server:

1. Change to the directory where the connector server was installed.
2. Unzip the Exchange connector ZIP file in the directory from Step1.

Note: If the AD connector DLL was updated by any one-off releases or patches, then do not update the AD DLL provided as part of the Exchange connector bundle zip file.

3. Restart the connector server service. Or, you can also restart the connector server using the following command:

```
ConnectorServer.exe /run
```

5.3.5 Postinstallation Tasks for the Exchange Connector

- Configuring the Connector Server in Oracle Waveset
- Updating to Newer Versions of the Connector-related Configuration in Oracle Waveset
- Creating an Exchange Connector Resource
- Editing the Exchange Connector Tabbed User Form

Configuring before and after actions for the Exchange connector is the same as for the Active Directory connector. See Configuring Before and After Actions for the Active Directory Connector.

5.3.5.1 Configuring the Connector Server in Oracle Waveset

Before you create an Exchange connector resource, configure the connector server in Oracle Waveset, as follows.

1. Click the Configure tab and then select the Connector Servers subtab.
2. Click **New** and provide the required configuration details, including:
   - Hostname or IP Address
   - Connector Server Port
   - Connector Server Key
   - Any user-specific name of the connector server
3. Click Save.

   If the connector server is up and running, the status is displayed as Available.

If the status is displayed as "Failed: Connection refused", start the connector server service or execute the following command:

```
ConnectorServer.exe /run
```

5.3.5.2 Updating to Newer Versions of the Connector-related Configuration in Oracle Waveset

To update Oracle Waveset with the newer versions of the Exchange connector artifacts:

1. Log in to the computer hosting Oracle Waveset and navigate to the Oracle Waveset installation directory.
2. Unzip the ExchangeConnector-idmglue-1.0.8.zip file in the installation directory.

   This will extract the following files:

   - WEB-INF/lib/ExchangeConnector-idmglue.jar
   - sample/connectors/ExchangeConnector-idmglue/migration.xml
   - sample/connectors/ExchangeConnector-idmglue/postProcess.xml
   - sample/connectors/ExchangeConnector-idmglue/resourceWizard.xml
   - sample/connectors/ExchangeConnector-idmglue/userForm.xml
3. Import the resourceWizard.xml file in the sample/connectors/ExchangeConnector-idmglue/ directory.
4. (Optional) Import the remaining XML files in the sample/connectors/ExchangeConnector-idmglue/ directory as required.

5.3.5.3 Creating an Exchange Connector Resource

To create an Exchange connector resource, follow these steps:

1. Log in to the Oracle Waveset Administrator interface.
2. Create the Exchange connector resource by following the Create Windows Exchange Connector Resource wizard.
3. Select the Exchange Connector Version as "2.0.0.1".
4. Select the connector server on which the Exchange connector bundle is deployed.
5. Specify values for the Exchange connector, depending on your deployment. For more information, see:

   - Exchange Connector Resource Configuration Parameters
   - Object Classes and Attributes Supported by the Exchange Connector

5.3.5.4 Editing the Exchange Connector Tabbed User Form

Edit the Tabbed User Form to use Exchange User Form for the Exchange resource created in Creating an Exchange Connector Resource as follows:

1. Go to the Oracle Waveset debug page:

   `http://`*host-name*`:`*port*`/idm/debug`
2. Select User Form from the drop-down box, which is adjacent to List Objects, and then click on List Objects.

3. Search for the Tabbed User Form and then click Edit. Under the _FM_ATTRIBUTES field:

   If this is the first connector using the dynamic form, search for the "MissingFields" string. Then, replace the `<FieldRef name='MissingFields'/>` block with following block:

   ```
   <FieldRef name='accountId'/>
   <FormRef name='Exchange User Form'>
   <Property name='RESOURCE_NAME' value='Exchange Server'/>
   </FormRef>
   ```

   If any other connector is configured already, then add the following block between `</FormRef>` and `</Field>`:

   ```
   <FormRef name='Exchange User Form'>
   <Property name='RESOURCE_NAME' value='Exchange Server'/>
   ```

   where Exchange Server is the resource name created in Creating an Exchange Connector Resource.

4. Navigate to the top of this form and add following line under the `<include>` tag:

   ```
   <ObjectRef type='UserForm' id='' name='Exchange User Form'/>
   ```

   The id will be picked up automatically.

5. Save the form.

5.4 Using the Exchange Connector

- Object Classes and Attributes Supported by the Exchange Connector
- Exchange Connector Sample Forms

5.4.1 Object Classes and Attributes Supported by the Exchange Connector

The Exchange connector support the following object classes:

- __ACCOUNT__ Object Class for the Exchange Connector
- __GROUP__ (or Group) Object Class for the Exchange Connector
- organizationalUnit Object Class for the Exchange Connector
- __MAILBOXDATABASE__ Object Class for the Exchange Connector
- __DISTRIBUTIONGROUP__ Object Class for the Exchange Connector

> **Note:** The Exchange connector extends the Active Directory connector schema. The Exchange connector supports any arbitrary Active Directory schema attribute used by Exchange Server and a predefined subset of Exchange PowerShell `cmdlet` parameters.

5.4.1.1 __ACCOUNT__ Object Class for the Exchange Connector

The following table lists only the Exchange Connector attributes that are additional to the Active Directory connector. For a complete list of the attributes, see __ACCOUNT__ Object Class for the Active Directory Connector.

Unless noted otherwise, an attribute is single-valued and optional, and can be created, updated, and read.

Table 5–5 Additional __ACCOUNT__ Attributes for the Exchange Connector

Attribute Name	Exchange Mapped Attribute	Type	Description
Database	Database	String	The list of available databases from the target is returned by default. When provisioning an Oracle Waveset user to an Exchange resource, the Database attribute value must be a distinguished name in the following format: CN=Mailbox Database 123456789,CN=Databases,CN=Exchange Administrative Group(ABCDEFG1234567),CN=Administrative Groups,CN=ABC Org,CN=Microsoft Exchange,CN=Services,CN=Configuration,DC=ABC,DC=COM Once set, this attribute is not updatable. Required for Exchange Server 2007 but optional for Exchange Server 2010.
ExternalEmailAddress	ExternalEmailAddress	String	Not returned by default. Required for both Exchange Server 2007 and Exchange Server 2010, if RecipientType is MailUser.
RecipientType	RecipientType	String	Required. Not returned by default.
Alias	Alias	String	Mailbox alias, which is generally the same as sAMAccountName. **Note:** sAMAccountName is the user login for Microsoft Active Directory.
SimpleDisplayName	SimpleDisplayName	String	Used to display an alternative description of the object when only a limited set of characters is permitted. This limited set of characters consists of ASCII characters 26 through 126, inclusively.
DisplayName	DisplayName	String	Name of a user as displayed in the address book. This is usually a combination of the user's first name, middle initial, and last name.
MaximumRecipients	RecipientLimits	String	Specifies the maximum number of recipients per message to which this mailbox can send.
MaxOutgoingMessageSize	MaxSendSize	String	Specifies the maximum size of messages that this mailbox can send.
MaxIncomingMessageSize	MaxReceiveSize	String	Specifies the maximum size of messages that this mailbox can receive.
UseStorageDefaults	UseDatabaseQuotaDefaults	Boolean	Specifies that this mailbox uses the quota attributes specified for the mailbox database where this mailbox resides.

Table 5–5 (Cont.) Additional __ACCOUNT__ Attributes for the Exchange Connector

Attribute Name	Exchange Mapped Attribute	Type	Description
ReceiptQuota	ProhibitSendReceiveQuota	String	Specifies the mailbox size at which the user associated with this mailbox can no longer send or receive messages.
TransmitQuota	ProhibitSendQuota	String	Specifies the mailbox size at which the user associated with this mailbox can no longer send messages.
MailboxWarningSize	IssueWarningQuota	String	Specifies the mailbox size at which a warning message is sent to the user.
ArchiveMailboxSize	ArchiveQuota	String	The archive mailbox size at which messages will no longer be accepted.
ArchiveMailboxWarningSize	ArchiveWarningQuota	String	The archive mailbox size at which a warning message is sent to the user.
RetainDeletedItems	UseDatabaseRetentionDefaults	Boolean	Specifies that this mailbox uses default values to handle deleted items or messages.
RetainDeletedItemsFor	RetainDeletedItemsFor	String	Specifies the length of time to keep deleted items.
RetainDeletedItemsUntilBackup	RetainDeletedItemsUntilBackup	Boolean	Specifies whether to retain deleted items until the next backup. The two possible values for this parameter are $true or $false.
HiddenFromAddressListsEnabled	HiddenFromAddressListsEnabled	Boolean	Specifies whether this mailbox is hidden from address lists. The two possible values for this parameter are $true or $false.
EmailAddressPolicyEnabled	EmailAddressPolicyEnabled	Boolean	Specifies whether the e-mail address policy for this mailbox is enabled. The two possible values for this parameter are $true or $false.
PrimarySMTPAddress	PrimarySMTPAddress	String	Specifies the address that external users see when they receive a message from this mailbox.
Distributiongroups	DistributionGroup	String	Multi-valued attribute.

Note: Oracle Waveset brings all the groups (Active Directory and Exchange distribution groups) from the target system for the first time and then caches it. From the next time the user form is opened, the groups are loaded from the cache. If a new group is added on the target system and if the group needs to be reflected in Oracle Waveset user form, then you must clear the cache. In addition, if you create a new resource, you must clear cache. To do so:

1. Go to the Oracle Waveset debug page at:

   `http://`*host_name*`:`*port*`/idm/debug`

2. Click **Clear Resource Object List Cache.**

   A message is displayed indicating that the Resource Object List cache has been cleared. Every time a new resource is created, you must clear the cache.

The following table lists the PowerShell `cmdlet` parameters supported by the Exchange connector. None of these attributes are returned by default.

The Exchange connector supports all multi-valued attributes that are supported by Exchange Server target. The attributes in the following table are supported for Exchange Server 2010, unless the description specifies Exchange Server 2007.

Note: By default, the Exchange connector includes only the DistributionGroup attribute in the userForm.xml file. To support additional multi-valued attributes, edit the userForm.xml and postProcess.xml files as described in Supporting Multi-Valued Attributes for the Exchange Connector.

Table 5–6 PowerShell `cmdlet` Parameters Supported by the Exchange Connector

Attribute Name	Exchange Mapped Attribute	Type	Description
AcceptMessagesOnlyFrom	AcceptMessagesOnlyFrom	String	Multi-valued.
AcceptMessagesOnlyFromDLMembers	AcceptMessagesOnlyFromDLMembers	String	Multi-valued.
AcceptMessagesOnlyFromSendersOrMembers	AcceptMessagesOnlyFromSendersOrMembers	String	Multi-valued.
BypassModerationFromSendersOrMembers	BypassModerationFromSendersOrMembers	String	Multi-valued.
CustomAttribute1 through CustomAttribute15	CustomAttribute1 through CustomAttribute15	String	
EmailAddresses	EmailAddresses	String	
ExtensionCustomAttribute1 through ExtensionCustomAttribute5	ExtensionCustomAttribute1 through ExtensionCustomAttribute5	String	Multi-valued. Attributes are supported only by Exchange Server 2010 SP2 and later releases.
Extensions	Extensions	String	Multi-valued. Supported for Exchange Server 2007.
GrantSendOnBehalfTo	GrantSendOnBehalfTo	String	Multi-valued.
ModeratedBy	ModeratedBy	String	Multi-valued.
RejectMessagesFrom	RejectMessagesFrom	String	Multi-valued.
RejectMessagesFromDLMembers	RejectMessagesFromDLMembers	String	Multi-valued.
RejectMessagesFromSendersOrMembers	RejectMessagesFromSendersOrMembers	String	Multi-valued.
RequireSenderAuthenticationEnabled	RequireSenderAuthenticationEnabled	Boolean	
WindowsEmailAddress	WindowsEmailAddress	String	

5.4.1.2 Supporting Multi-Valued Attributes for the Exchange Connector

By default, the Exchange connector includes only the multi-valued DistributionGroup attribute. To support additional multi-valued attributes, edit the userForm.xml and postProcess.xml files, as follows:

1. In the postProcess.xml file, add mappings for the multi-valued attributes that you want to add. For example, the mapping for the GrantSendOnBehalfTo attribute is:

```
<MapEntry
key='accountAttributes[mapName==GrantSendOnBehalfTo].attributeName'
value='GrantSendOnBehalfTo'/>
<MapEntry key='accountAttributes[mapName==GrantSendOnBehalfTo].multi'
value='true'/>
```

2. In the userForm.xml file, add the Field reference for the multi-valued attributes you want to add. For example, the reference for the GrantSendOnBehalfTo attribute is:

```
<Field name='accounts[$(RESOURCE_NAME)].GrantSendOnBehalfTo'>
<Display class="TextArea">
<Property name="title" value="SendOnBehalf" />
<Property name="rows" value="4" />
<Property name="columns" value="60" />
<Property name="format" value="list" />
<Property name="sorted" value="true" />
</Display>
</Field>
```

3. Import the postProcess.xml and userForm.xml files into Oracle Waveset.

5.4.1.3 __GROUP__ (or Group) Object Class for the Exchange Connector

The __GROUP__ (or Group) object class represents Active Directory groups excluding distribution groups (which are still part of Active Directory groups but are specific to Exchange).

For a list of the __GROUP__ (or Group) object class attributes, see __GROUP__ (Group) Object Class for the Active Directory Connector.

5.4.1.4 organizationalUnit Object Class for the Exchange Connector

For a list of the organizationalUnit object class attributes, see organizationalUnit Object Class for the Active Directory Connector.

5.4.1.5 __MAILBOXDATABASE__ Object Class for the Exchange Connector

The __MAILBOXDATABASE__ object class is added to the existing Exchange connector schema to make it more convenient to retrieve mail store and mailbox database data from each version of the target system.

Table 5–7 __MAILBOXDATABASE__ Object Class for the Exchange Connector

Attribute Name	Type	Description
__NAME__	String	Required.

5.4.1.6 __DISTRIBUTIONGROUP__ Object Class for the Exchange Connector

There are no additional attributes to the schema in the Exchange connector for groups; however the connector forces the group type to be a Distribution group.

5.4.2 Exchange Connector Sample Forms

The Exchange User Form (UserForm.xml) is provided with the Exchange connector.

In addition, the following forms are also provided: PostProcess.xml, Migration.xml, and ResourceWizard.xml

5.5 Frequently Asked Questions (FAQs)

You can refer the following FAQs as guidelines and to troubleshoot connector issues. The following topics are discussed in this section:

- FAQs Common to Both Exchange 2010 and 2007
- FAQs Related to Exchange 2010
- FAQs Related to Exchange 2007

5.5.1 FAQs Common to Both Exchange 2010 and 2007

The following are FAQs on connector issues common to both Exchange 2010 and Exchange 2007:

1. What is the recommended system configuration for the computer hosting and running the connector server?

   Answer: The computer on which you want to install and run the connector server must meet the following requirements:

   - The computer hosting the connector server must have Intel Dual-Core Processor, 2 GHz with 4 GB RAM or a computer with similar configuration. If you have a computer dedicated to the connector server, then 2 GB RAM is sufficient.
   - Microsoft Windows Server 2003 or 2008, either 32-bit or 64-bit versions.

2. Where should I install the connector server for the Exchange connector?

   Answer: Install the connector server on a computer that belongs to the same domain as that of the target Exchange server.

3. Why cannot I see the log files corresponding to the connector operations in the computer hosting Oracle Identity Manager?

   Answer: The Exchange connector uses the built-in logging mechanism of the .NET framework. Therefore, all connector logs are generated on the computer hosting the connector server. See Enabling Logging for more information.

4. After extracting the contents of the connector bundle into the *CONNECTOR_SERVER_HOME* directory, I observed some DLLs. Does it matter whether the computer hosting the connector server is 32-bit or 64-bit?

   Answer: No, you can use the same DLLs on both 32-bit and 64-bit computers.

5. Can a single connector server be used to deploy the Active Directory User Management connector bundle and the Exchange connector bundle?

   Answer: Yes, a single connector server can host both the Active Directory User Management and the Exchange connector bundles.

   While deploying the Exchange connector, ensure not to replace the existing ActiveDirectory.Connector.dll file on the connector server.

6. Does the Exchange connector support managing only Active Directory user as well? If so, what steps need to be done?

   Answer: Yes, the Exchange connector supports managing Active Directory users as well. To manage Active Directory users, in the user form select User in the recipient type drop down box. In this case, only Active Directory user will be created.

   Alternatively, you can use Oracle Waveset Connector for Microsoft Active Directory to manage Active Directory users. For more information, see Oracle Waveset Connector for Microsoft Active Directory.

5.5.2 FAQs Related to Exchange 2010

The following are FAQs on connector issues specific to Exchange 2010:

1. In what format should the IT resource parameter ExchangeUser be specified?

   Answer: It should be in the *DOMAIN_NAME\USER_NAME* format.

2. How do I ensure that the username and password provided in the IT resource are correct?

   Answer: Follow the steps mentioned in Table 5–9, " Troubleshooting Connector Issues with Exchange 2010" for the error "unknown user name or bad password."

3. What is the minimum permission/role that the user provided in IT resource should have?

   Answer: The user should be part of the Recipient Management group.

4. What are Exchange 2010 specific requirements that must be met by the computer hosting connector server?

   Answer: The host computer should meet all the prerequisites of Remote PowerShell. For more information, see the topic on Connect Remote Exchange Management Shell to an Exchange Server at:

   `http://technet.microsoft.com/en-in/library/dd297932%28v=exchg.141%29.aspx`

5. Does the computer hosting the connector server need to have Exchange Management Tools installed?

   Answer: No.

5.5.3 FAQs Related to Exchange 2007

The following are FAQs on connector issues specific to Exchange 2007:

1. Does the connector support RTM version of Exchange 2007?

   Answer: No.

2. What values do I have to provide for ExchangeUser, ExchangeUserPassword, and ExchangeServerHost in the Exchange IT resource?

   Answer: No values are required for these attributes. You can leave them blank. As the connector communicates to Exchange 2007 via local runspace, the connector does not use username or password provided in IT resource to connect to Exchange server. It uses the username and password of the user who starts the connector server.

3. Does the computer hosting the connector server need to have Exchange Management Tools installed?

   Answer: Yes.

4. What is the minimum permission/role of the user who starts the connector server?

   Answer:

   - User should be part of the Exchange Recipient Administrators group.
   - User should be part of the Account Operators group in the domain where the distribution group exists.

5. What are Exchange 2007 specific requirements that need to be met by the computer hosting the connector server?

   Answer: The host computer needs to have Exchange Management Tools installed.

5.6 Troubleshooting Connector Issues

The following tables list solutions to some issues associated with the Exchange connector:

- Table 5–8, " Troubleshooting Common Connector Issues"
- Table 5–9, " Troubleshooting Connector Issues with Exchange 2010"
- Table 5–10, " Troubleshooting Connector Issues with Exchange 2007"

Table 5–8 lists solutions to some commonly encountered issues associated with the Exchange connector:

Table 5–8 Troubleshooting Common Connector Issues

Problem Description	Solution
The Exchange connector throws the following error: `Could not find domain controller for user <user_name>`	The connector tries to get the domain controller where the Active Directory (AD) user was created using the value provided in the Directory Adminstrator's Account field. This value must be same as the value provided for the User Principal Name field during AD provisioning. If there is any mismatch, the connector throws this error. Ensure the values provided for these two fields are same.
The Exchange connector does not log any information. Logging is enabled for the connector in ConnectorServer.exe.Config file. The line <add name="ExchangeSwitch" value="4" /> has been added and connector server has been restarted.	Ensure the log file location and name as specified in the ConnectorServer.exe.Config file is valid. Also, ensure the user who is running the connector server has write permission on the log file. Then, restart the connector server.

The Exchange connector throws the following error while updating a user:
`"Account not found in Resource"`
	Multiple domain controllers may be configured for the same domain. In such a case, add a field to the ResourceWizard.xml file as follows:
	`<Field name="resourceAttributes[SyncDomainController].value" required="false">`
	`<Display class="Text">`
	`<Property name="title" value="SyncDomainController"/>`
	`<Property name="help" value="SyncDomainController"/>`
	`</Display>`
	`</Field>`
	Then, import the XML file and specify the domain controller (host) value in the resource form.

Table 5–9 lists solution to a commonly encountered issue associated with the connector when using Exchange 2010:

Table 5–9 Troubleshooting Connector Issues with Exchange 2010

Problem Description	Solution
The Exchange connector throws the following error: `ConnectorServer.exe Error: 0 : System.Management.Automation.Remoting.PSRemotingTransportException: Connecting to remote server failed with the following error message : Logon failure: unknown user name or bad password. For more information, see the about_Remote_Troubleshooting Help topic.`	Ensure the username and password specified are correct. The username must be in the format DomainName\UserName. User distinguished name (DN) must *not* be mentioned as a value for the ExchangeUser resource parameter. If this does not solve the issue, verify if you can connect to Exchange Server from the computer hosting the connector server using a remote PowerShell window using the same credentials by following below commands: `$cred = Get-Credential` //provide same credentials as specified in the resource `$Session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri http://<ExchangeServerHostName>/PowerShell/ -Authentication Kerberos -Credential $cred` //provide same Exchange Server host name as provided in the resource parameter `Import-PSSession $session` //this should import Exchange cmdlets without any issues. Ideally by this step, you should encounter the issue. `Remove-PSSession -Session $Session` // remove newly created test session If the above steps complete without any error, then check Windows event logs for more information. **Alternate Solution:** Run the **Enable-PSRemoting** cmdlet to configure the Exchange Server computer to receive Windows PowerShell remote commands that are sent by using the WS-Management technology. For more information about the Enable-PSRemoting cmdlet, see: `http://technet.microsoft.com/en-us/library/hh849694.aspx`

Table 5–10 lists solution to commonly encountered issue associated with the connector when using Exchange 2007:

Table 5–10 Troubleshooting Connector Issues with Exchange 2007

Problem Description	Solution
The Exchange connector throws the following error while adding a user to a distribution group: `ConnectorServer.exe Error: 0 : Org.IdentityConnectors.Framework.Common.Exceptions.ConnectorException: Problem while PowerShell execution Org.IdentityConnectors.Framework.Common.Exceptions.ConnectorException: Active Directory operation failed on MachineName.connectordevroot1.com. This error is not retriable. Additional information: Insufficient access rights to perform the operation.`  `Active directory response: 00002098: SecErr: DSID-03150BB9, problem 4003 (INSUFF_ACCESS_RIGHTS)`	For Exchange 2007, the service account must be a member of the Exchange Recipient Administrator role and the Account Operator role in every domain where the distribution group exists. Add the user to the Account Operator role of the domain where the distribution group exists and restart the connector server.

6

Oracle Waveset Connector for Google Apps

This chapter includes the following information about the Google Apps connector for Oracle Waveset:

- About the Google Apps Connector
- Deploying the Google Apps Connector
- Using the Google Apps Connector
- Troubleshooting the Google Apps Connector

6.1 About the Google Apps Connector

- Overview of the Google Apps Connector
- Security Considerations for the Google Apps Connector
- Certified Components for the Google Apps Connector
- Supported Languages for the Google Apps Connector

6.1.1 Overview of the Google Apps Connector

The Google Apps connector supports provisioning of accounts and groups to Google Apps. For information about Google Apps, see `http://www.google.com/apps/`.

The Google Apps connector is implemented using the Identity Connector Framework (ICF). The ICF provides a container that separates the connector bundle from the application. The ICF also provides common features that developers would otherwise need to implement on their own, such as connection pooling, buffering, time outs, and filtering. For more information about the ICF, see Chapter 1, "Identity Connectors Overview".

The Google Apps connector is new to Oracle Waveset, and there is not a corresponding resource adapter.

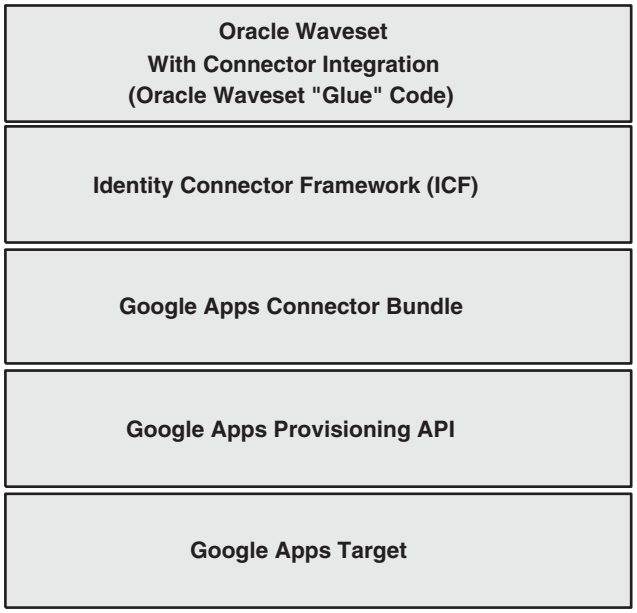
This section provides the following additional information about the Google Apps connector:

- Google Apps Connector Architecture
- Google Apps Connector Features
- Configuration Properties for the Google Apps Connector
- Resource Object Management for the Google Apps Connector

6.1.1.1 Google Apps Connector Architecture

The following figure shows the Google Apps connector architecture.

Figure 6–1 Google Apps Connector Architecture



The Google Apps connector architecture includes these components:

- Oracle Waveset includes the connector integration files. These files are XML files that provide the configuration information necessary to transform data from a resource to Oracle Waveset. Integration files are sometimes called the connector "glue" code.
- The Identity Connector Framework (ICF) provides a container that separates the connector bundle from the application. The ICF also provides common features that developers would otherwise need to implement on their own, such as connection pooling, buffering, time outs, and filtering.
- The Google Apps connector bundle uses the Google Apps Provisioning API to access the Google Apps target system. For the specific helper libraries used to talk to this API, see Certified Components for the Google Apps Connector.

6.1.1.2 Google Apps Connector Features

The Google Apps connector supports these provisioning operations:

- Create account
- Update account
- Delete account
- Enable/disable account
- Update password
- Full reconciliation
- Filtering
- Agentless target deployment

6.1.1.3 Configuration Properties for the Google Apps Connector

The following table describes the configuration parameters for the Google Apps connector.

Table 6–1 Configuration Properties for the Google Apps Connector

Property	Description
Google Apps Domain Admin URL	URL for the Google Apps domain. The default value of `https://www.google.com/a/feeds/mydomain.com` must be changed.
Domain	Name of your Google Apps domain. The default value of `mydomain.com` must be changed.
Login	Administrator account for the Google Apps domain. This administrator must have rights to create and manage users. The name should not include the @domain component.
Password	Administrator password.
Proxy Host	Proxy host name. Specify this field when the connector is to be used in the network protected by a web proxy. Consult with your network administrator for more information about proxy configuration.
Proxy Port	Port of the web proxy.
Proxy Username	Account name to use for the proxy.
Proxy Password	Password for the account specified in the Proxy Username field.

6.1.1.4 Resource Object Management for the Google Apps Connector

Oracle Waveset manages the following Google Apps objects:

Table 6–2 Resource Object Management for the Google Apps Connector

ResourceObject	Supported Features	Attributes Managed
Account (`__ACCOUNT__` object class)	Create, update, delete, enable/disable, full reconciliation	__NAME__, familyName, givenName, quota, nicknames, groups, __PASSWORD__, __ENABLE__, isAdmin, changePasswordAtNextLogin
Group (`__GROUP__` object class)	Create, update, delete	__NAME__, groupName, groupDescription, groupPermissions, owners, members

Note: The `__NAME__` attribute is not updatable. For more information, see Object Classes and Attributes Supported by the Google Apps Connector.

6.1.2 Security Considerations for the Google Apps Connector

This section provides the following information:

- Supported Connections for the Google Apps Connector
- Required Administrator Privileges for the Google Apps Connector

6.1.2.1 Supported Connections for the Google Apps Connector

The Google Apps connector supports the `HTTPS` protocol.

Note: The Google Apps Connector uses the Google Apps Provisioning API to talk to Google Apps. The HTTPS protocol is used to communicate with the Google Apps Provisioning API web services. Depending on your application server configuration, you might need to import Google certificates to your application server keystore or truststore. Appropriate certificates can be extracted from the following URLs:

- `https://www.google.com/a/feeds/`*`yourdomain`*`/user/2.0/`
- `https://www.google.com/a/feeds/`*`yourdomain`*`/nickname/2.0/`
- `https://apps-apis.google.com/a/feeds/group/2.0/`

In the first two URLs, *`yourdomain`* represents your specific domain.

6.1.2.2 Required Administrator Privileges for the Google Apps Connector

The user name that connects to Google Apps must be able to create, edit, and delete accounts and groups.

6.1.3 Certified Components for the Google Apps Connector

The Google Apps connector for Oracle Waveset is certified with the following components:

Table 6–3 Certified Components for the Google Apps Connector

Component	Requirement
Oracle Waveset	Oracle Waveset 8.1.1 Patch 6
Identity Connector Framework (ICF)	ICF 1.1 or later
Google Apps	Google Data Java Client 1.33 and Google Collections 1.0-rc1

6.1.4 Supported Languages for the Google Apps Connector

The Google Apps connector is localized in the following languages:

- Arabic
- Chinese (Simplified)
- Chinese (Traditional)
- Danish
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazilian)
- Spanish

6.2 Deploying the Google Apps Connector

Deploying the Google Apps requires the following tasks:

- Installing the Google Apps Connector
- Creating a Google Apps Connector Resource

6.2.1 Installing the Google Apps Connector

To install the Google Apps connector, you must have access to the file system on the application server.

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the Google Apps Connector.
2. Download the Google Data Java Client 1.33 and Google Collections 1.0-rc1 from the following locations:
   - Google Data Java Client 1.33 at `http://code.google.com/p/gdata-java-client`.

     `gdata-appsforyourdomain-1.0.jar`

     `gdata-client-1.0.jar`

     `gdata-core-1.0.jar`
   - Google Collections 1.0-rc1 at `http://code.google.com/p/google-collections/`.

     `google-collect-1.0-rc1.jar`

   To find a specific library, on each page, click the Downloads tab, select All downloads in the Search drop-down menu, and then click Search. Download the appropriate version of the library ZIP file and then extract and use the JAR files listed above.
3. Stop the Oracle Waveset web application.
4. Copy the Google Apps JAR files from Step 2 to the *`InstallDir`*`/WEB-INF/lib` directory on the application server.
5. Start the Oracle Waveset web application.
6. Log in to the Oracle Waveset Administrator interface and select the Google Apps connector.

6.2.2 Creating a Google Apps Connector Resource

To create a Google Apps connector resource in Oracle Waveset, follow these steps:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the Google Apps Connector.
2. Log in to the Oracle Waveset Administrator interface.
3. Create the Google Apps connector resource by following the Create Google Apps Connector Resource wizard.
4. Specify values for the configuration parameters, as described in Configuration Properties for the Google Apps Connector.

For additional information about creating resources, see "Understanding and Managing Waveset Resources" in the *Oracle Waveset 8.1.1 Business Administrator's Guide*.

6.3 Using the Google Apps Connector

This section provides information related to using the Google Apps connector, including:

- Object Classes and Attributes Supported by the Google Apps Connector
- Sample Forms for the Google Apps Connector

6.3.1 Object Classes and Attributes Supported by the Google Apps Connector

The Google Apps connector for Oracle Waveset supports the following object classes:

- __ACCOUNT__ Object Class for the Google Apps Connector
- __GROUP__ Object Class for the Google Apps Connector
- Attribute Mapping Changes

6.3.1.1 __ACCOUNT__ Object Class for the Google Apps Connector

The Google Apps connector supports the __ACCOUNT__ object class (Google Apps User) and the attributes shown in the following table. Unless noted in the description, an attribute is creatable, updatable, readable, and returned by default.

Table 6–4 __ACCOUNT__ Object Class for the Google Apps Connector

Attribute Name	Type	Required	Description
__NAME__	String	Yes	User's account name. Not updatable.
familyName	String	Yes	User's last name.
givenName	String	Yes	User's first name.
quota	Integer	No	Disk space in megabytes (MB) allocated for this user. **Note**: The default value is 25 GB for each user account. This field is not updatable. To set user account quotas, a domain must have a Google agreement.
nicknames	String	No	Other names this user is known by. Can be multi-valued. Not returned by default.
groups	String	No	Groups this user is a member of. Can be multi-valued. Not returned by default.
__PASSWORD__	GuardedString	Yes	User's password. Not readable and not returned by default.
__ENABLE__	Boolean	No	If set to true, enables this user.
isAdmin	Boolean	No	If set to true, allows this user to be assigned admin privileges. Default is false.
changePasswordAtNextLogin	Boolean	No	If set to true, forces this user to change his or her password at the next login. Default is false.

6.3.1.2 __GROUP__ Object Class for the Google Apps Connector

The Google Apps connector supports the __GROUP__ object class (Google Apps Group) and the attributes shown in the following table. Unless noted in the description, an attribute is creatable, updatable, readable, and returned by default.

Table 6–5 __GROUP__ Object Class for the Google Apps Connector

Attribute Name	Type	Required	Description
__NAME__	String	Yes	Not updatable.
groupName	String	Yes	Name of this group.
groupDescription	String	Yes	Description of this group.
groupPermissions	String	Yes	Permissions for this group.
owners	String	No	Owners of this group. Can be multi-valued. Not returned by default.
members	String	No	Members of this group. Can be multi-valued. Not returned by default.

6.3.1.3 Attribute Mapping Changes

Post processing (`postProcess.xml`) changes the attribute mapping shown in the following table. The other Google Apps connector attributes are mapped to the Oracle Waveset attributes with the same names.

Table 6–6 Attribute Mapping Changes

Oracle Waveset Attribute	Type	Google Apps Connector Attribute	Type
firstname	String	givenName	String
lastname	String	familyName	String

6.3.2 Sample Forms for the Google Apps Connector

The Google Apps connector for Oracle Waveset provides the following sample forms, located in the `sample/connectors/googleapps-idmglue` directory:

- `userForm.xml`
- `groupCreate.xml`
- `groupUpdate.xml`

After you install the Google Apps connector, the sample forms usually requires some modification, depending on your deployment.

For example, to support the Google Apps sample user form, modify the Tabbed User Form as follows:

1. Go to Oracle Waveset debug page:

   `http://`*`host_name`*`:`*`port`*`/idm/debug`

2. Select User Form from the drop-down box, which is adjacent to List Objects, and then click on List Objects.
3. Search for the Tabbed User Form and then click Edit.
4. Make the following changes in the Tabbed User Form:
   a. Add the Google Apps sample user form inside the `<Include>` tag, as follows:

```
<Include>
...
<ObjectRef type='UserForm' name='Google Apps IdC User Form'/>
</Include>
```

b. Add the following <FormRef...> element before the <FormRef name='MissingFields'/> tag:

```
<FormRef name='Google Apps IdC User Form'>
    <Property name='RESOURCE_NAME' value='GoogleApps'/>
</FormRef>
```

c. In the <FormRef...> element you added in the previous step, set the RESOURCE_NAME property value to the name of the specific Google Apps resource.

6.4 Troubleshooting the Google Apps Connector

Use the Oracle Waveset debug pages to set trace options on the org.identityconnectors.googleapps.* or org.identityconnectors.* packages.

If you want to narrow the scope of the trace, use one or more of the following classes, listed in order or priority:

- org.identityconnectors.googleapps.GoogleAppsConnector
- org.identityconnectors.googleapps.GoogleAppsUserOps
- org.identityconnectors.googleapps.GoogleAppsGroupOps
- org.identityconnectors.googleapps.GoogleAppsClient

7

Oracle Waveset Connector for PeopleSoft Employee Reconciliation

This chapter includes the following information about the PeopleSoft Employee Reconciliation connector for Oracle Waveset:

7.1 About the PeopleSoft Employee Reconciliation Connector

7.1.1 Overview of the PeopleSoft Employee Reconciliation Connector

The PeopleSoft Employee Reconciliation connector for Oracle Waveset supports reconciliation with PeopleSoft target systems. The connector uses PeopleTools with the PeopleSoft Human Resource Management System (HRMS).

The PeopleSoft Employee Reconciliation connector reconciliation is based on the "pull" model, which uses Active Sync to pull, or fetch, data from the PeopleSoft target system and then update the respective data in Oracle Waveset. The "pull" model requires customization on the PeopleSoft target system, which includes writing new component interfaces to pull the data from the target system.

The PeopleSoft Employee Reconciliation connector does not support the "push" model for reconciliation. Therefore, you cannot use this connector to create or modify PeopleSoft target system accounts.

Also, to create a User Profile in Oracle Waveset, the PeopleSoft Employee Reconciliation connector requires the Employee Job assignment. This limitation is based on the custom component interface configurations from the resource adapter.

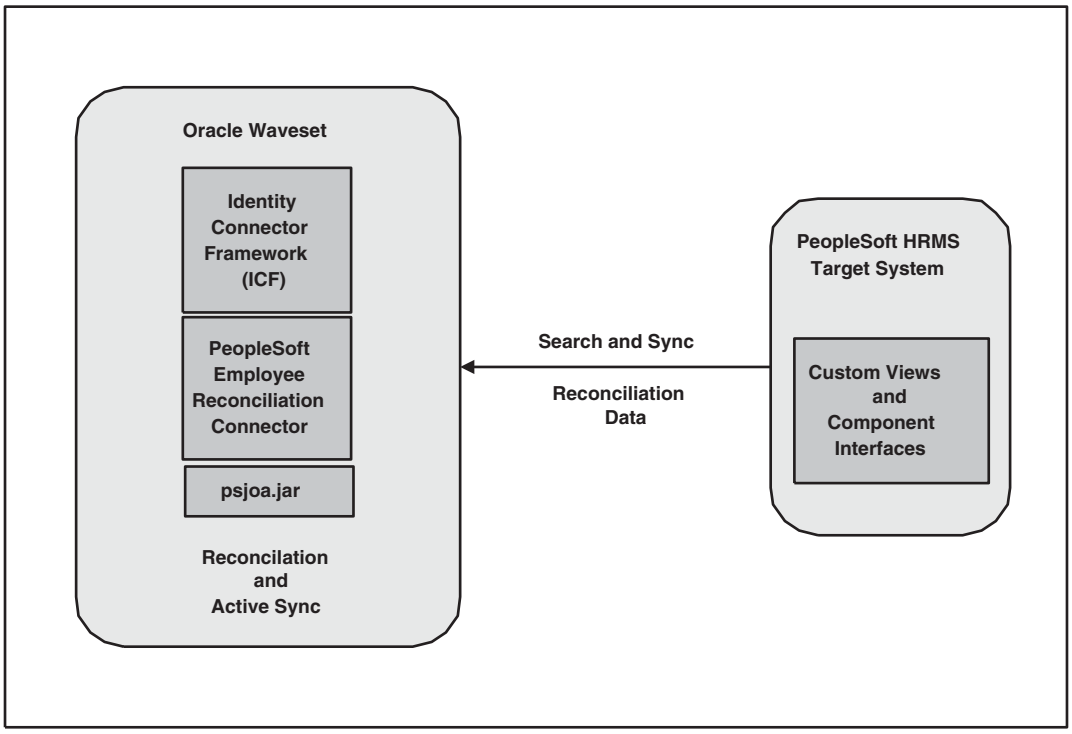
This section provides the following information about the PeopleSoft Employee Reconciliation connector:

- PeopleSoft Employee Reconciliation Connector Architecture
- PeopleSoft Employee Reconciliation Connector Features
- PeopleSoft Employee Reconciliation Connector Configuration Properties

7.1.1.1 PeopleSoft Employee Reconciliation Connector Architecture

The following figure shows the component architecture for the PeopleSoft Employee Reconciliation connector.

Figure 7–1 PeopleSoft Employee Reconciliation Connector Architecture



The PeopleSoft Employee Reconciliation connector architecture includes these components:

- Oracle Waveset includes the connector integration files. These files are XML files that provide the configuration information necessary to transform data from a resource to Oracle Waveset. Integration files are sometimes called the connector "glue" code.
- The Identity Connector Framework (ICF) provides basic provisioning, logging, and other functions that Oracle Waveset (and Oracle Identity Manager) connectors can use.
- The PeopleSoft Employee Reconciliation connector uses custom views and component interfaces to fetch the data from the PeopleSoft HRMS target system.

  The connector requires the PeopleSoft Java Object Adapter (`psjoa.jar` file) to access the target system data. The version of the `psjoa.jar` must match the version of the installed PeopleSoft target system.

If you are installing the PeopleSoft Employee Reconciliation connector in the Connector Server, see also Installing the PeopleSoft Employee Reconciliation Connector in the Connector Server.

7.1.1.2 PeopleSoft Employee Reconciliation Connector Features

The PeopleSoft Employee Reconciliation connector supports reconciliation operations as follows:

- Full reconciliation: Initially, all Employee accounts are fetched from the PeopleSoft target system, and resources (Waveset users) are created in Oracle Waveset.
- Active Sync (incremental reconciliation): Updated data or newly created Employee account records on the PeopleSoft target system are fetched from target system and Oracle Waveset resources (Waveset users) are updated or created, respectively, using this new information.

The PeopleSoft Employee Reconciliation connector does **not** support the following operations:

- Create, update, delete, enable, disable, or rename accounts on the PeopleSoft target system
- Password update
- Pass-through authentication
- Before and after actions

7.1.1.3 PeopleSoft Employee Reconciliation Connector Configuration Properties

The following table describes the configuration properties for the PeopleSoft Employee Reconciliation connector.

Table 7–1 PeopleSoft Employee Reconciliation Connector Configuration Properties

Property	Description
Host	Hostname or IP address of the PeopleSoft target resource.
TCP Port	Port number on which the PeopleSoft target resource is listening.
User	User ID of a PeopleSoft user with the permissions required to invoke methods on the component interfaces.
Password	User's password.

For information about setting these properties, see Postinstallation Tasks for the PeopleSoft Employee Reconciliation Connector.

7.1.2 Security Considerations for the PeopleSoft Employee Reconciliation Connector

This section describes the following security considerations:

- Supported Connections for the PeopleSoft Employee Reconciliation Connector
- Required Administrative Privileges for the PeopleSoft Employee Reconciliation Connector

7.1.2.1 Supported Connections for the PeopleSoft Employee Reconciliation Connector

Oracle Waveset uses Oracle Jolt to communicate with the PeopleSoft Employee Reconciliation connector.

7.1.2.2 Required Administrative Privileges for the PeopleSoft Employee Reconciliation Connector

The administrative user that connects must have permissions on the `LH_AUDIT_EFFDT_COMP_INTF` and `LH_EMPLOYEE_COMP_INTF` component interfaces. For more information, see Step 6: Configure PeopleTools.

7.1.3 Certified Components for the PeopleSoft Employee Reconciliation Connector

The PeopleSoft Employee Reconciliation connector for Oracle Waveset is certified with the following components:

Table 7–2 Certified Components for the PeopleSoft Employee Reconciliation Connector

Component	Requirement
Oracle Waveset	Oracle Waveset 8.1.1 Patch 6
Target systems	PeopleSoft HRMS 8.9, 9.0, 9.1, and 9.2
PeopleTools	PeopleTools 8.48, 8.49, 8.50, 8.51, and 8.53
Identity Connector Framework (ICF)	ICF 1.0 or later
JDK	JDK 1.5 or later If you are using PeopleTools 8.53, see the note in Section 7.3.3, "Installing the PeopleSoft Employee Reconciliation Connector in the Connector Server" for information related to JDK requirement.

7.1.4 Supported Languages for the PeopleSoft Employee Reconciliation Connector

The PeopleSoft Employee Reconciliation connector is localized in the following languages:

- Arabic
- Chinese (Simplified and Traditional)
- Danish
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (European and Brazilian)
- Spanish

7.2 Migrating to the PeopleSoft Employee Reconciliation Connector From a Resource Adapter

If you currently have the PeopleSoft Component resource adapter deployed, this section describes how to migrate the adapter to the PeopleSoft Employee Reconciliation connector.

To migrate from a PeopleSoft Component resource adapter, follow these steps:

1. Make sure you have installed the Oracle Waveset patch shown in Certified Components for the PeopleSoft Employee Reconciliation Connector.
2. Log in to the Oracle Waveset Administrator interface.
3. Select the Resources tab and then the Migrate Adapters tab.
4. Follow the Migration Wizard and provide the values for the PeopleSoft Employee Reconciliation connector.

   The Migrate Adapters operation automatically migrates the PeopleSoft Component resource adapter to the PeopleSoft Employee Reconciliation connector.

7.3 Deploying the PeopleSoft Employee Reconciliation Connector

You can deploy the PeopleSoft Employee Reconciliation connector either locally in Oracle Waveset or remotely in the Connector Server, as described in the following sections:

- Preinstallation Tasks for the PeopleSoft Employee Reconciliation Connector
- Installing the PeopleSoft Employee Reconciliation Connector in Oracle Waveset
- Installing the PeopleSoft Employee Reconciliation Connector in the Connector Server
- Postinstallation Tasks for the PeopleSoft Employee Reconciliation Connector

7.3.1 Preinstallation Tasks for the PeopleSoft Employee Reconciliation Connector

To integrate the PeopleSoft Employee Reconciliation connector with Oracle Waveset, use the following PeopleSoft tools:

- Application Designer. Use this tool to build and configure the Oracle Waveset project.
- PeopleTools browser-based application. Use this tool to configure component interfaces, roles, and user profiles.

To configure PeopleSoft for use with Oracle Waveset, follow these steps:

- Step 1: Create a New Project
- Step 2: Edit the Oracle Waveset Objects
- Step 3: Build the Project
- Step 4: Manually Execute the `audittrigger` Script
- Step 5: Enable Auditing
- Step 6: Configure PeopleTools
- Step 7: Testing Component Interface

- Step 8: Prune the Audit Log

7.3.1.1 Step 1: Create a New Project

To create a new project with the PeopleSoft Application Designer, follow these steps:

1. Create a new project in the Application Designer by selecting the **File -> New** menu. Then select Project from the list.
2. Name the project by performing a save. Use the **File -> Save Project As...** menu, and enter a unique name for the project such as Waveset.
3. Create the objects within the project by performing the tasks in Step 2: Edit the Oracle Waveset Objects.

7.3.1.2 Step 2: Edit the Oracle Waveset Objects

The Oracle Waveset project contains the following types of objects:

- Fields
- Records
- Pages
- Components
- Component Interfaces

Create these objects within the Application Designer. Each of these objects is described in detail in the following sections.

7.3.1.2.1 Fields Create the following fields:

- AUDIT_PROC_ORDER. Set the field type to Character and set the length to 20.
- AUDIT_PROC_END. Set the field type to Character and set the length to 20.
- AUDIT_PROC_DATE. Set the field type to Date

The following procedure describes how to create the AUDIT_PROC_ORDER field. Use this procedure as a guide to create any field needed in your project.

To create the AUDIT_PROC_ORDER field, follow these steps:

1. Select **File -> New... -> Field**.
2. Select Character field type.
3. Set the field length to 20.
4. Assign the Label ID AUDIT_PROC_ORDER.
5. Save the field by selecting **File -> Save**. Assign it the name AUDIT_PROC_ORDER.
6. Select **Insert -> Current Definition** to add the field to the project.

Create all the fields you need for this project. You might need to drag these fields into one or more records.

7.3.1.2.2 Records There are three records (two views and one table) that must be defined within the Application Designer. The following record descriptions illustrate a typical implementation. The records can be customized to the needs of the implementation by adding or changing fields.

AUDIT_EFFDT_LH View

The AUDIT_EFFDT_LH view is polled by the connector. Oracle Waveset uses the following fields to query for events that have not yet been processed:

- AUDIT_PROC_ORDER. This field must specify the Key, Search Key, List Box Item, and From Search Field keys.
- AUDIT_PROC_END. This field must specify the Key, Search Key, List Box Item, and Through Search Field fields.
- EMPLID and EMPL_RCD. These are required non-key properties that are used by an Oracle Waveset query to fetch employee data.

All other fields in the AUDIT_EFFDT_LH table are optional.

The following table describes the Use Display characteristics of the AUDIT_EFFDT_LH view:

Table 7–3 Use Display Characteristics of the AUDIT_EFFDT_LH View

Field Name	Type	Key	Ordr	Dir	Srch	List	Sys	Default
AUDIT_PROC_ORDER	Char	Key	1	Asc	Yes	Yes	No	-
AUDIT_PROC_END	Char	Key	-	Asc	Yes	Yes	No	-
AUDIT_STAMP	DtTm	-	-	-	No	No	No	-
EFFDT	Date	-	-	-	No	No	No	%date
AUDIT_OPRID	Char	-	-	-	No	No	No	-
AUDIT_ACTN	Char	-	-	-	No	No	No	-
AUDIT_RECNAME	Char	-	-	-	No	No	No	-
EMPLID	Char	-	-	-	No	No	No	'NEW'
EMPL_RCD	Nbr	-	-	-	No	No	No	-

Note: Ordering of the fields for AUDIT_EFFDT_LH view should match with the columns of the SQL query used below.

Information in the last audit entry is stored in Oracle Waveset as a "lastProcessed" configuration object to be used (and updated) on subsequent searches of the AUDIT_EFFDT_LH view. Maintenance of the `lastProcessed` Configuration object by the PeopleSoft Employee Reconciliation connector prevents records from being processed more than once.

Example 7–1 SQL Code to Generate the AUDIT_EFFDT_LH View

The following SQL code is used to generate the AUDIT_EFFDT_LH view:

```
SELECT audit1.AUDIT_PROC_ORDER AS AUDIT_PROC_ORDER
,audit1.AUDIT_PROC_ORDER AS AUDIT_PROC_END
,audit1.AUDIT_STAMP AS AUDIT_STAMP
,audit1.EFFDT AS EFFDT
,audit1.AUDIT_OPRID AS AUDIT_OPRID
,audit1.AUDIT_ACTN AS AUDIT_ACTN
,audit1.AUDIT_RECNAME AS AUDIT_RECNAME
,audit1.EMPLID AS EMPLID
,CAST(audit1.EMPL_RCD AS INTEGER) AS EMPL_RCD FROM PS_AUDIT_PRS_DATA audit1
WHERE audit1.AUDIT_PROC_DATE <= %CurrentDateIn
```

```
AND NOT EXISTS (
SELECT * FROM PS_AUDIT_PRS_DATA audit2
WHERE audit2.AUDIT_PROC_DATE <= %CurrentDateIn
AND audit2.AUDIT_PROC_ORDER > audit1.AUDIT_PROC_ORDER
AND (audit2.EMPLID = audit1.EMPLID AND audit2.EMPL_RCD = audit1.EMPL_RCD) );
```

The final line in this SQL code sample prevents Oracle Waveset from seeing operations with effective dates until the effective date has arrived.

AUDIT_PRS_DATA Table

The AUDIT_PRS_DATA table must contain the following fields:

- AUDIT_PROC_ORDER. This field must specify the Key, Search Key, List Box Item, and From Search field keys. In addition, this field must be set to Required so that PeopleSoft puts a non-null integrity constraint on the database column.
- AUDIT_PROC_DATE. This field must specify the Alternate Search Key, List Box Item. In addition, this field must be set to Required so that PeopleSoft puts a non-null integrity constraint on the database column.
- EMPLID and EMPL_RCD. These are required non-key properties that are used by an Oracle Waveset query to fetch employee data.

All other fields in the AUDIT_PRS_DATA table are optional.

The following table describes the Use Display characteristics of the AUDIT_PRS_DATA view:

Table 7–4 Use Display Characteristics of the AUDIT_PRS_DATA View

Field Name	Type	Key	Ordr	Dir	Srch	List	Sys	Default
AUDIT_PROC_ORDER	Char	Key	1	Asc	Yes	Yes	No	
AUDIT_PROC_DATE	Date	Alt		Asc	No	No	No	
AUDIT_STAMP	DtTm				No	No	No	%date
AUDIT_OPRID	Char				No	No	No	'ANON'
AUDIT_ACTN	Char				No	No	No	'C'
AUDIT_RECNAME	Char				No	No	No	'ANON'
EMPLID	Char				No	No	No	'NEW'
EFFDT	Date				No	No	No	%date
EMPL_RCD	Nbr				No	No	No	

PERS_SRCH_LH View

The PERS_SRCH_LH view must contain the EMPLID and EMPL_RCD fields, with the Key, Search Key, and List Box Item keys selected. All other fields provide the data that is synchronized with Oracle Waveset. It is up to the PeopleSoft Active Sync form to map this data into the Oracle Waveset user account.

The following table describes the Use Display characteristics of the PERS_SRCH_LH view:

Table 7–5 Use Display Characteristics of the PERS_SRCH_LH View

Field Name	Type	Key	Ordr	Dir	Srch	List	Sys
EMPLID	Char	Key	1	Asc	Yes	Yes	No
EMPL_RCD	Nbr	Key	2	Asc	Yes	Yes	No
NAME	Char				No	Yes	No
LAST_NAME_SRCH	Char				No	Yes	No
SETID_DEPT	Char				No	Yes	No
DEPTID	Char				No	Yes	No
ADDRESS1	Char				No	Yes	No
EMPL_STATUS	Char				No	Yes	No
FIRST_NAME	Char				No	Yes	No
LAST_NAME	Char				No	Yes	No
MIDDLE_NAME	Char				No	Yes	No
REPORTS_TO	Char				No	Yes	No
JOBCODE	Char				No	Yes	No
COMPANY	Char				No	Yes	No
NAME_INITIALS	Char				No	Yes	No
COUNTRY	Char				No	Yes	No
PHONE	Char				No	Yes	No
CITY	Char				No	Yes	No
STATE	Char				No	Yes	No
POSTAL	Char				No	Yes	No

Note: Ensure that the order of the fields in the PERS_SRCH_LH view are same as the order of the fields in pers_srch_lh.sql (SQL query used for this view).

The following SQL code is used to generate the PERS_SRCH_LH view.

Note: For your convenience, the `peoplesoft/idm.zip` file on the installation media contains an SQL script file named `pers_srch_lh.sql` that duplicates the following SQL code.

Example 7–2 SQL Code to Generate the PERS_SRCH_LH View

```
SELECT P.EMPLID
 ,A.EMPL_RCD
 ,P.NAME
 ,P.LAST_NAME_SRCH
 ,A.SETID_DEPT
 ,A.DEPTID
 ,P.ADDRESS1
 ,A.EMPL_STATUS
 ,P.FIRST_NAME
 ,P.LAST_NAME
```

```
,P.MIDDLE_NAME
,A.REPORTS_TO
,A.JOBCODE
,A.COMPANY
,P.NAME_INITIALS
,P.COUNTRY
,P.PHONE
,P.CITY
,P.STATE
,P.POSTAL
FROM PS_Job A
, PS_PERSONAL_DATA P
WHERE A.EMPLID = P.EMPLID
AND A.EffDt = (
SELECT MAX(C.EffDt)
FROM PS_Job C
WHERE C.EmplID = A.EmplID
AND C.EMPL_RCD = A.EMPL_RCD
AND C.EffDt <= %CurrentDateIn)
AND A.EffSeq = (
SELECT MAX(D.EffSeq)
FROM PS_Job D
WHERE D.EmplID = A.EmplID
AND D.EMPL_RCD = A.EMPL_RCD
AND D.EffDt = A.EffDt)
```

The WHERE clause returns the current employee record for the given employee ID. PeopleSoft allows multiple records for a given employee, each of which has its own effective date/effective sequence. This clause returns the record whose effective date/effective sequence pair is the latest out of all those that are already effective (whose effective date has occurred).

The WHERE clause returns null for an employee whose sunrise date is in the future.

If the record does not work while performing reconciliation, use the following workaround:

1. Back up the EMPLMT SRCH_All by including it in a new project. Select **Tools -> Copy to file**.
2. Open the emplmt_srch_all view then rename it to PERS_SRCH_LH. Select **File -> Rename**. Then configure PERS_SRCH_LH view as described.
3. Restore the backup by selecting **Tools -> Copy project from file**.

7.3.1.2.3 Pages

The Oracle Waveset project must also contain the following pages for the component interface only:

- LH_AUDIT_EFFDT
- LH_EMPLOYEE_DATA

Drag and drop the required records into the page definitions.

LH_AUDIT_EFFDT

The LH_AUDIT_EFFDT page contains fields defined in the AUDT_EFFDT_LH table. This page is not displayed on the PeopleSoft GUI. Therefore, the layout and ordering of the fields is not important.

The following table describes the Use Display characteristics of the LH_AUDIT_EFFDT page. All items are defined in the AUDT_EFFDT_LH record.

Table 7–6 Use Display Characteristics of the LH_AUDIT_EFFDT Page

Label	Type	Field
Unique order to process	Edit Box	AUDIT_PROC_ORDER
EmplID	Edit Box	EMPLID
Upper bound for search	Edit Box	AUDIT_PROC_END
Empl Rcd Nbr	Edit Box	EMPL_RCD
Date and Time Stamp	Edit Box	AUDIT_STAMP
Effective Date	Edit Box	EFFDT
User ID	Edit Box	AUDIT_OPRID
Action	Drop Down List	AUDIT_ACTN
Audit Record Name	Edit Box	AUDIT_RECNAME

LH_EMPLOYEE_DATA

The LH_EMPLOYEE_DATA page is the container for the fields defined in the PERS_SRCH_LH view. All items are defined in the PERS_SRCH_LH record.

The following table describes the Use Display characteristics of the LH_EMPLOYEE_DATA page:

Table 7–7 Use Display Characteristics of the LH_EMPLOYEE_DATA Page

Label	Type	Field
EmplID	Edit Box	EMPLID
Name	Edit Box	NAME
Last Name	Edit Box	LAST_NAME_SRCH
Department SetID	Edit Box	SETID_DEPT
Department	Edit Box	DEPTID
Address Line 1	Edit Box	ADDRESS1
Personnel Status	Edit Box	PER_STATUS
Employee Status	Edit Box	EMPL_STATUS
First Name	Edit Box	FIRST_NAME
Last Name	Edit Box	LAST_NAME
Middle Name	Edit Box	MIDDLE_NAME
Reports To Position	Edit Box	REPORTS_TO
Job Code	Edit Box	JOBCODE
Company	Edit Box	COMPANY
Name Initials	Edit Box	NAME_INITIALS
Country	Edit Box	COUNTRY
Telephone	Edit Box	PHONE
City	Edit Box	CITY

Table 7–7 (Cont.) Use Display Characteristics of the LH_EMPLOYEE_DATA Page

Label	Type	Field
State	Edit Box	STATE
Postal Code	Edit Box	POSTAL
Empl Rcd Nbr	Edit Box	EMPL_RCD

7.3.1.2.4 Components Components are the bridge between pages and menus. Once you have created your pages, you add them to one or more components to use them on menus or in business processes. Create a separate component for the each of the following pages:

- LH_AUDIT_EFFDT
- LH_EMPLOYEE_DATA

The default component names are LH_AUDIT_EFFDT and LH_EMPLOYEE_COMP.

To create the LH_AUDIT_EFFDT component, follow these steps:

1. Select **File -> New... -> Component**.
2. Select **Insert -> Page Into Component...**. Specify the name as LH_AUDIT_EFFDT.
3. Select **File -> Definition/Object Properties**. Then go to Use and Search Record AUDIT_EFFDT_LH.
4. **Select File -> Save** and name the component LH_AUDIT_EFFDT.
5. Select **Insert -> Component** to add the component to the project.

On the LH_EMPLOYEE_DATA component, set the Search Record as PERS_SRCH_LH.

7.3.1.2.5 Component Interfaces A component interface is a PeopleTools object that exposes a PeopleSoft component for synchronous access from another application, such as Oracle Waveset. Create a separate component interface for each component you created. The default names for the component interfaces are `LH_AUDIT_EFFDT_COMP_INTF` and `LH_EMPLOYEE_COMP_INTF`. These values can be modified on the General Active Sync Settings page of the Active Sync Wizard.

To create the `LH_AUDIT_EFFDT_COMP_INTF` component interface, follow these steps:

1. Select **File -> New... -> Component Interface**.
2. Specify a source component, such as LH_AUDIT_EFFDT. When prompted, select **Yes**.
3. Select **File -> Save**. Specify the name LH_AUDIT_EFFDT_COMP_INTF.
4. Select **Insert -> Component Interface** to add the component interface to the project.

Note:

- Add all the audit related employee data fields as properties to `LH_AUDIT_EFFDT_COMP_INTF` component interface.
- Create `LH_EMPLOYEE_COMP_INTF` component interface by following steps through 1 to 4 and add all the employee data fields as properties to `LH_EMPLOYEE_COMP_INTF` component interface.

7.3.1.3 Step 3: Build the Project

Use this procedure to build the project and create PeopleSoft views and tables in the database.

To build the project using the Application Designer, follow these steps:

1. Select **Build -> Project**. The Build dialog appears.
2. In the Build Options area, select the Create Tables and Create Views options. In the Build Execute Options area, select the Execute SQL now option.
3. Click Settings. The Build Settings dialog appears.
4. Verify that the Recreate table if it already exists option is selected.
5. Click the Logging tab.
6. In the Logging Level area, select the Fatal errors, warnings and information messages option.
7. In the Logging Output area, enter a unique log file name.
8. Click OK, and then click Build to build the project and to create views and tables.

   Application Designer might display a warning message similar to the following message:

   ```
   Potentially data destructive settings are active. Continue
   the build process?
   ```

9. Click **Yes** to continue to build process.

Note: After importing and building the project, you must test the components in Application Designer. The reliability of the import project feature within PeopleSoft varies from release to release. Therefore, validation of the objects is very important.

7.3.1.4 Step 4: Manually Execute the `audittrigger` Script

The `idm.zip` file on the installation media contains an Oracle SQL script named `audittrigger.oracle`. This script creates the trigger and sequence necessary to maintain the `AUDIT_PROC_DATE` and `AUDIT_PROC_ORDER` columns of the `PS_AUDIT_PRS_DATA` table.

Note: The `audittrigger.oracle` script is available only for Oracle. If you are using a different database, convert the script to run on that database.

The `audittrigger.oracle` script or its equivalent must be run every time you rebuild the PeopleSoft project.

7.3.1.5 Step 5: Enable Auditing

From the Application Designer, enable auditing on the JOB and PERSONAL_DATA tables, and possibly on the POSITION_DATA and EMPLOYMENT tables. This is record-level auditing that writes a simple summary record with the operator and the EMPLID of the changed record.

To enable auditing, follow these steps:

1. Launch the Application Designer.
2. Select **File -> Open** to display the Open Object dialog.
3. Select Record from the Object type menu, and then type JOB in the Name field.
4. Click Open to open the record.
5. Select **File -> Properties** to open the record properties, and then click the Use tab.
6. In the Record Name field, select AUDIT_PRS_DATA.
7. In the Audit Options area, select the Add, Change, and Delete options. Leave the Selective option unchecked.

Repeat these steps for the PERSONAL_DATA table and other tables that will be triggers for data synchronization.

For more information, see "Creating Record Definitions" in the Application Designer documentation.

7.3.1.6 Step 6: Configure PeopleTools

To complete the configuration process, use the PeopleTools browser-based GUI to assign component interfaces to a permission list, create a role and assign permission lists to the role, and assign the role to user profiles. Refer to the PeopleTools documentation for more information about these entities. This section describes these tasks:

- Authorizing a Component Interface
- Assigning a PeopleSoft Role to the Component Interfaces
- Assigning a Role to a User Profile

7.3.1.6.1 Authorizing a Component Interface To authorize a component interface, follow these steps:

1. Log in to the PeopleTools browser-based GUI and navigate to **Home -> People Tools -> Maintain Security -> Use -> Permission Lists**.

   For Peoplesoft 9.x, this path is **Home -> People Tools -> Security -> Permissions & Roles -> Permission List**.
2. Select the **Add a New Value** link and enter a value such as LH_ALL
3. Click on the right arrow in the tabs section near the top of the page until the Component Interface tab is displayed. Then click on the Component Interface tab.
4. Enter an existing component interface, such as `LH_AUDIT_EFFDT_COMP_INTF`, in the text box.
5. Click the Edit link to go to the Component Interface Permissions page.

6. Click the Full Access button to enable full access for all the methods, or use the drop-down menus to assign access for individual methods. Click OK to return to the Permission Lists page.
7. Click the + (plus) button. An additional text box will be displayed.
8. Enter a different existing component interface, such as `LH_EMPLOYEE_COMP_INTF`, in the text box.
9. Repeat steps 5 and 6.
10. Save your changes.

7.3.1.6.2 Assigning a PeopleSoft Role to the Component Interfaces To assign a PeopleSoft Role to a component interface, follow these steps:

1. Navigate to **Home -> People Tools -> Maintain Security -> Use -> Roles**.

   For Peoplesoft 9.x, the path is **Home -> People Tools -> Security -> Permissions & Roles -> Roles**.
2. Select the Add a New Value link and enter a value such as LH_ROLE.
3. Click the Permission Lists tab.
4. Enter an existing Permission List, such as LH_ALL.
5. Save your changes.

7.3.1.6.3 Assigning a Role to a User Profile To assign a role to a user profile, follow these steps:

1. Navigate to **Home -> People Tools -> Maintain Security -> Use -> User Profiles**.

   For Peoplesoft 9.x, the path is **Home -> People Tools -> Security -> User Profiles -> User Profiles**.
2. Enter an existing user ID. This user can be specified as the user on the Resource Parameters page in Oracle Waveset.

   Note: You can also create a new user. Refer to the PeopleSoft documentation for more information about the requirements of a user account.
3. Select the Roles tab.
4. Click the + (plus) button. An additional text box will be displayed.
5. Enter the name of a role, such as LH_ROLE.
6. Save your changes.

7.3.1.7 Step 7: Testing Component Interface

To test the `LH_AUDIT_EFFDT_COMP_INTF` and `LH_EMPLOYEE_COMP_INTF` component interfaces, follow these steps:

1. Open the Application Designer.
2. Select **File -> Open.**
3. In the definition, select **Component Interface.**
4. In the Name field, enter `LH_EMPLOYEE_COMP_INTF`.

5. Click **Open.**
6. Select the Component Interface and right click **Test Component Interface.**

   A new window opens.
7. In the Find Keys field, specify a value for EMPLID.
8. Click **Find.**
9. Ensure that the values for all the fields appear correctly.

> **Note:** Repeat steps 1 through 9 for `LH_AUDIT_EFFDT_COMP_INTF` component interface and ensure that the values for all the fields are appearing correctly after creating the employee records.

7.3.1.8 Step 8: Prune the Audit Log

Oracle Waveset does not delete audit events from the audit log. The PeopleSoft administrator must set up a task to prune old audit entries. This task must retain transactions with a future effective date until Oracle Waveset processes them. That is, entries whose `AUDIT_PROC_DATE` is in the future must **not** be pruned.

7.3.2 Installing the PeopleSoft Employee Reconciliation Connector in Oracle Waveset

To install the PeopleSoft Employee Reconciliation connector in Oracle Waveset, you must have access to the file system on the application server. You must also have Oracle Waveset administrator privileges.

To install the PeopleSoft Employee Reconciliation connector in Oracle Waveset, follow these steps:

1. Make sure you have installed the Oracle Waveset patch shown in Certified Components for the PeopleSoft Employee Reconciliation Connector.
2. Stop the Oracle Waveset web application.
3. Explode the `idm.war` file.
4. Copy the PeopleSoft Employee Reconciliation connector bundle JAR file (`org.identityconnectors.peoplesoftintfc-`*`version`*`.jar`) to the `$WSHOME/WEB-INF/bundles` directory of the Oracle Waveset web application.

   In this step, *`version`* represents the connector bundle version. For example: `org.identityconnectors.peoplesoftintfc-1.0.5963.jar`
5. Copy the `peoplesoftcomponent-idmglue.jar` file from the PeopleSoft Employee Reconciliation connector `glue code/WEB-INF/lib` directory to the `$WSHOME/WEB-INF/lib` directory.
6. Copy the following XML files from the PeopleSoft Employee Reconciliation connector `glue code/sample/connectors` directory to the `$WSHOME/sample/connectors/peoplesoftcomponent-idmglue` directory:
   - `postProcess.xml`
   - `resourceWizard.xml`
   - `migration.xml`
7. Copy the `psjoa.jar` file from the PeopleSoft installation media to the `$WSHOME/WEB-INF/lib` directory. The version of the `psjoa.jar` must match the version of the installed PeopleSoft target system.

Note: You can also add the `psjoa.jar` file to the PeopleSoft Employee Reconciliation connector bundle `/lib` directory, which can enable support for different versions of the PeopleSoft target system. For example, to add the `psjoa.jar` file to the connector bundle:

```
jar uvf
org.identityconnectors.peoplesoftcomponent-1.0.596
3.jar lib/psjoa.jar
```

8. Recreate the `idm.war` file.
9. Start the Oracle Waveset web application.
10. When you set identity system parameters, specify `Employee ID` for the display name attribute.

For more information, see Postinstallation Tasks for the PeopleSoft Employee Reconciliation Connector.

7.3.3 Installing the PeopleSoft Employee Reconciliation Connector in the Connector Server

The Connector Server requires a JDK to run. For the JDK requirements, see Certified Components for the PeopleSoft Employee Reconciliation Connector. If necessary, set your `JAVA_HOME` environment variable to point to your specific installation.'

Note: If you are using PeopleTools 8.53, following is the JDK requirement:

- If you are already using a Connector Server, then it is mandatory to use JDK 1.7.0_02 as the minimum version in the Connector Server.
- If the you are not using Connector Server and Oracle Waveset is not using JDK 1.7.0_02, then follow one of the below mentioned steps:
  - Refer the certification matrix and upgrade the JDK version used by Oracle Waveset to JDK 1.7.0_02 if it is supported.
  - If JDK 1.7.0_02 is not supported for Oracle Waveset, then it is mandatory to use a Connector Server with minimum JDK 1.7.0_02. In addition, create a new Connector Server Definition for the Connector Server and specify it in the PeopleSoft Component Connector Resource.

This section describes the following subsections:

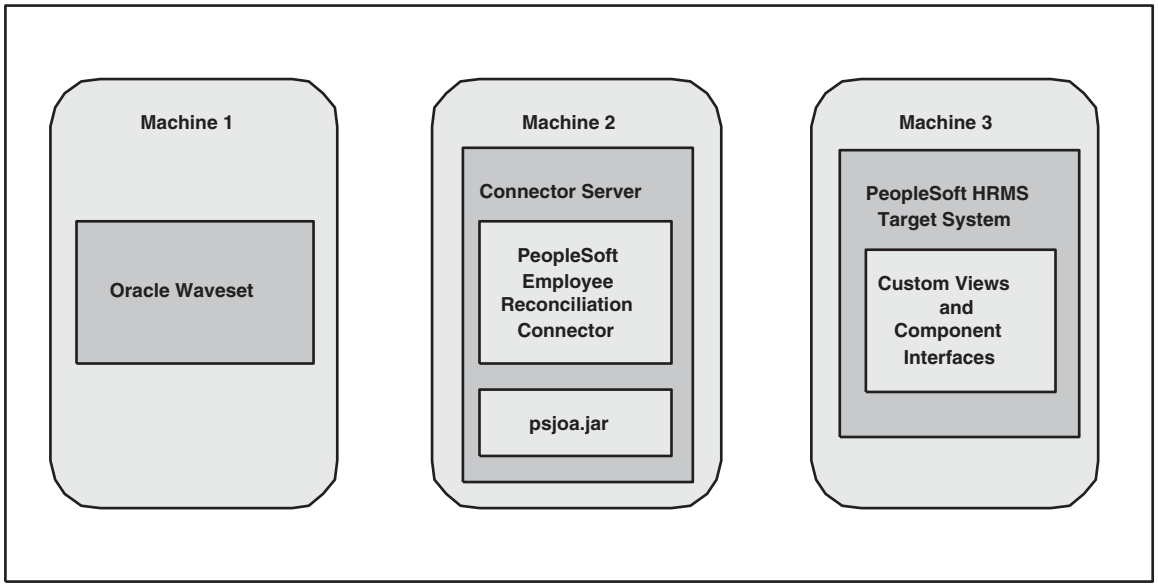
- PeopleSoft Employee Reconciliation Connector Deployment Architecture With the Connector Server
- Installing and Configuring the Connector Server
- Running the Connector Server on Windows Systems

- Running the Connector Server on UNIX and Linux Systems
- Installing the PeopleSoft Employee Reconciliation Connector in the Connector Server

7.3.3.1 PeopleSoft Employee Reconciliation Connector Deployment Architecture With the Connector Server

If you install the PeopleSoft Employee Reconciliation connector in the Connector Server, the following figure shows the distributed deployment architecture.

Figure 7–2 PeopleSoft Employee Reconciliation Connector Deployment Architecture With the Connector Server



A PeopleSoft Employee Reconciliation connector deployment with the Connector Server includes these components:

- **Machine 1** has Oracle Waveset deployed.
- **Machine 2** has the PeopleSoft Employee Reconciliation connector installed in the Connector Server. The Connector Server is part of the Identity Connector Framework (ICF).

  The PeopleSoft Employee Reconciliation connector is installed in the `CONNECTOR_SERVER_HOME/bundles` directory.

  The appropriate PeopleSoft Java Object Adapter (`psjoa.jar`) file is installed in the `CONNECTOR_SERVER_HOME/lib` directory. The version of the `psjoa.jar` must match the version of the installed PeopleSoft target system.
- **Machine 3** has the PeopleSoft HRMS target system deployed.

7.3.3.2 Installing and Configuring the Connector Server

To install and configure the Connector Server, follow these steps:

1. Create a new directory on the machine where you want to install the Connector Server. In this section, `CONNECTOR_SERVER_HOME` represents this directory.

2. Unzip the Connector Server package in your new directory from Step 1. The Connector Server package is available with the Identity Connector Framework (ICF).
3. In the `ConnectorServer.properties` file, set the following properties, as required by your deployment. The `ConnectorServer.properties` file is located in the `conf` directory.

Property	Description
`connectorserver.port`	Port on which the Connector Server listens for requests. The default is 8759.
`connectorserver.bundleDir`	Directory where the connector bundles are deployed. The default is `bundles`.
`connectorserver.libDir`	Directory in which to place dependent libraries. The default is `lib`.
`connectorserver.usessl`	If set to `true`, the Connector Server uses SSL for secure communication. The default is `false`. If you specify `true`, use the following options on the command line when you start the Connector Server: ■ `-Djavax.net.ssl.keyStore` ■ `-Djavax.net.ssl.keyStoreType` (optional) ■ `-Djavax.net.ssl.keyStorePassword`
`connectorserver.ifaddress`	Bind address. To set this property, uncomment it in the file (if necessary). The bind address can be useful if there are more NICs installed on the machine.
`connectorserver.key`	Connector Server key.

4. Set the properties in the `ConnectorServer.properties` file, as follows:
   - To set `connectorserver.key`, run the Connector Server with the `setKey` option.
   - For all other properties, edit the `ConnectorServer.properties` file manually.
5. The `conf` directory also contains the `logging.properties` file, which you can edit if required by your deployment.

7.3.3.3 Running the Connector Server on Windows Systems

To run the Connector Server on Windows systems, use the `ConnectorServer.bat` script as follows:

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

   The `ConnectorServer.bat` script supports the following options:

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

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

7.3.3.4 Running the Connector Server on UNIX and Linux Systems

To run the Connector Server on UNIX and Linux systems, use the `connectorserver.sh` script, as follows:

1. Make sure that you have set the properties required by your deployment in the `ConnectorServer.properties` file, as described in Installing and Configuring the Connector Server.
2. Change to the `CONNECTOR_SERVER_HOME/bin` directory.
3. Use the `chmod` command to set the permissions to make the `connectorserver.sh` script executable.
4. Run the `connectorserver.sh` script. The script supports the following options:

Option	Description
`/run [ -Jjava-option ]`	Runs the Connector Server in the console. Optionally, you can specify one or more Java options. For example, to run the Connector Server with SSL: `./connectorserver.sh /run -J-Djavax.net.ssl.keyStore=mykeystore.jks -J-Djavax.net.ssl.keyStorePassword=password`
`/start [ -Jjava-option ]`	Runs the Connector Server in the background. Optionally, you can specify one or more Java options.
`/stop`	Stops the Connector Server, waiting up to 5 seconds for the process to end.
`/stop n`	Stops the Connector Server, waiting up to *n* seconds for the process to end.

Option	Description
`/stop -force`	Stops the Connector Server. Waits up to 5 seconds and then uses the `kill -KILL` command, if the process is still running.
`/stop n -force`	Stops the Connector Server. Waits up to *n* seconds and then uses the `kill -KILL` command, if the process is still running.
`/setKey key`	Sets the Connector Server key. The `connectorserver.sh` script stores the hashed value of *key* in the `connectorserver.key` property in the `ConnectorServer.properties` file.

7.3.3.5 Installing the PeopleSoft Employee Reconciliation Connector in the Connector Server

To install the PeopleSoft Employee Reconciliation connector for Oracle Waveset in the Connector Server, follow these steps:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the PeopleSoft Employee Reconciliation Connector.
2. Stop the Connector Server.
3. Copy the PeopleSoft Employee Reconciliation connector bundle to the `CONNECTOR_SERVER_HOME/bundles` directory.
4. Copy the `psjoa.jar` file to the `CONNECTOR_SERVER_HOME/lib` directory.

   The version of the `psjoa.jar` must match the version of the installed PeopleSoft target system.
5. Start the Connector Server.

For information about starting and stopping the Connector Server, see Running the Connector Server on Windows Systems or Running the Connector Server on UNIX and Linux Systems.

7.3.4 Postinstallation Tasks for the PeopleSoft Employee Reconciliation Connector

Oracle Waveset requires the following additional configuration before you can create a new PeopleSoft Employee Reconciliation connector resource:

1. Start the Oracle Waveset web application.
2. Log in to the Oracle Waveset Administrator interface.
3. Load the following files into Oracle Waveset from the **Configure -> Import Exchange File** page:
   - `$WSHOME/sample/lib/peoplesoftcomponent-idmglue/migration.xml`
   - `$WSHOME/sample/lib/peoplesoftcomponent-idmglue/postProcess.xml`
   - `$WSHOME/sample/lib/peoplesoftcomponent-idmglue/resourceWizard.xml`
   - `$WSHOME/sample/forms/PeopleSoftForm.xml`
4. Create the PeopleSoft Employee Reconciliation connector resource. Configure the Resource Parameters page by completing the following fields.
   a. **Host** — Enter the hostname or IP address of the PeopleSoft target resource.
   b. **Port**— Enter the port number on which the PeopleSoft target resource is listening.

c. **User**— Enter the user ID of a PeopleSoft user with the permissions required to invoke methods on the component interfaces.

d. **Password**— Enter the password for User.

5. Edit the Synchronization Policy for the connector by configuring the following fields:

   - **Audit Component Interface Name** — Enter the component interface name to access the audit data. This component is used to search the AUDIT_PRS_DATA record in PeopleSoft.
   - Select the **Use Legacy Get Update Rows** checkbox, if required.
   - **Employee Component Interface Name** — Enter the component interface name to access Employee records managed by PeopleSoft. This component is used to Access the JOB and PERSONAL_DATA records in PeopleSoft.

   In addition, map the input form to the PeopleSoft Active Sync form.

6. Edit the Reconciliation Policy as follows:

   - **MISSING** — Create resource account for user
   - **UNASSIGNED** — Link resource account to user
   - **UNMATCHED** — Create new user based on resource account
   - **DISPUTED** — Delete resource account

7. On the General Active Sync Settings page of the Active Sync Wizard, specify the **Audit Component Interface Name** and the **Employee Component Interface Name**.

7.4 Using the PeopleSoft Employee Reconciliation Connector

This section provides information related to using the PeopleSoft Employee Reconciliation connector, including:

- Account Attributes for the PeopleSoft Employee Reconciliation Connector
- Sample Forms for the PeopleSoft Employee Reconciliation Connector

7.4.1 Account Attributes for the PeopleSoft Employee Reconciliation Connector

The following table provides information about thePeopleSoft Employee Reconciliation resource account attributes.

Note: During reconciliation, the PeopleSoft Employee Reconciliation connector displays only the `Employee ID`, `firstname`, and `lastname` of the users that are fetched from the PeopleSoft target system. To display additional employee information, add other supported attributes in `formlib.xml` and then map the attributes to the respective resource user attributes.

See Configuring Account Attributes for the PeopleSoft Employee Reconciliation Connector.

Table 7–8 Account Attributes for the PeopleSoft Employee Reconciliation Connector

Identity System Account Attribute	Resource Account Attribute	Description
accountId	EMPLID	Required.
ACTION	ACTION	Action code of up to 3 characters.
ACTION_REASON	ACTION_REASON	Reason code of up to 3 characters.
AUDIT_ACTN	AUDIT_ACTN	Type of action the system audited (A=add, C=change, D=delete).
AUDIT_OPRID	AUDIT_OPRID	Operator who caused the system to trigger the audit.
AUDIT_STAMP	AUDIT_STAMP	Date and time stamp.
AUDIT_RECNAME	AUDIT_RECNAMEN	Name of the record the system audited.
EFFSEQ	EFFSEQ	Effective sequence.
EFFDT	EFFDT	Effective date.
Employee ID	EMPL_ID	Key field used to uniquely identify users.
fullname	NAME	User's full name.
firstname	FIRST_NAME	User's first name.
lastname	LAST_NAME	User's last name.
Middle Name	MIDDLE_NAME	User's middle name
PS_PER_STATUS	PER_STATUS	Personnel status, such as employee or non-employee.
PS_EMPL_STATUS	EMPL_STATUS	Status of the employee, such as Active, Suspended, or Terminated.
Home Address	ADDRESS1	User's home address.
Department	DEPTID	User's department.
Manager	REPORTS_TO	User's manager.
Job Title	JOBCODE	Code that identifies the user's job title.
Initials	NAME_INITIALS	User's initials.
Country	COUNTRY	3-letter country code.
Company	COMPANY	Company name.
Home Phone	PHONE	User's home phone number.
Home City	CITY	City in which the user resides.
Home State	STATE	State in which the user resides.
Home Zip	POSTAL	User's home Zip or postal code.

7.4.1.1 Configuring Account Attributes for the PeopleSoft Employee Reconciliation Connector

During reconciliation, the PeopleSoft Employee Reconciliation connector displays only the `Employee ID`, `firstname`, and `lastname` of the users that are fetched from the PeopleSoft target system.

The following attributes are available in Oracle Waveset and can be mapped to the target system attributes to display complete employee information that is fetched from the PeopleSoft target system during reconciliation.

- address1
- city
- company
- country
- deptid (Department ID)
- emplstatus (Employee Status)
- jobcode
- phone
- postal (Postal Code)
- state

These attributes are available in the following file:

`$WSHOME/sample/formlib.xml`

To add additional attributes, follow these steps:

1. Stop the server where Oracle Waveset is deployed. For example, stop the Oracle Waveset deployment on Oracle WebLogic Server.
2. Add attributes to the `formlib.xml` file:
   a. Copy the `formlib.xml` file from `$WSHOME/sample`.
   b. Add a new attribute. For example, to add the Job Code, under the `<Field name="IdentityContent">` tag, add the following lines:

```
<Field name="global.jobcode">
<Display class="Text">
<Property name="title" value="Job Code"/>
</Display>
</Field>
```

   Repeat these steps for all additional attributes you want to add.
3. Start the Oracle Waveset web application.
4. Log in to the Oracle Waveset Administrator interface.
5. Click the Configure tab.
6. Click Import Exchange File and then import the `formlib.xml` file.
7. Click the Resources tab, the PeopleSoft Employee Reconciliation connector resource object (for example, PSFTER), and then Next.
8. Map each desired Identity System User Attribute to the Resource User Attribute.

   The drop-down menu on the left lists the attributes that are declared in the `formlib.xml` file, including any new attributes you added.
9. When you are finished mapping the attributes, click Next.

7.4.2 Sample Forms for the PeopleSoft Employee Reconciliation Connector

The PeopleSoft Employee Reconciliation connector does not include a sample form.

See Configuring Account Attributes for the PeopleSoft Employee Reconciliation Connector.

7.5 Troubleshooting the PeopleSoft Employee Reconciliation Connector

Use the Oracle Waveset debug pages to set trace options on the following class:

`org.identityconnectors.peoplesoftcomponent`

To set the logging options for the Connector Server, edit the properties in the `CONNECTOR_SERVER_HOME/conf/logging.properties` file. See Installing and Configuring the Connector Server.

For more information about enabling Oracle Waveset tracing and setting the tracing levels, see "Tracing Waveset Objects and Activities," in the *Oracle Waveset 8.1.1 System Administrator's Guide* in the following library:

`http://docs.oracle.com/cd/E19225-01/index.html`

8

Oracle Waveset Connector for PeopleSoft User Management

This chapter includes the following information about the PeopleSoft User Management connector for Oracle Waveset:

8.1 About the PeopleSoft User Management Connector

8.1.1 Overview of the PeopleSoft User Management Connector

The PeopleSoft User Management connector manages data in PeopleSoft through component interfaces. The connector can also manage additional PeopleSoft applications (such as HR and Financials) if these applications are installed on a system with a supported version of PeopleTools.

The PeopleSoft User Management connector is implemented using the Identity Connector Framework (ICF). The ICF provides a container that separates the connector bundle from the application. The ICF also provides common features that developers would otherwise need to implement on their own, such as connection pooling, buffering, time outs, and filtering. For more information about the ICF, see Chapter 1, "Identity Connectors Overview".

The PeopleSoft User Management connector is configured by default to support the USER_PROFILE, ROLE_MAINT, and DELETE_USER_PROFILE component interfaces. The connector can also use custom component interfaces to create, read, and update account data if the component interface supports the create, get, find, and

save methods. To delete accounts, the custom component interface must support get and save methods.

The PeopleSoft User Management connector supersedes the Peoplesoft Component Interface resource adapter. To migrate from a resource adapter deployment, see Migrating to the PeopleSoft User Management Connector from a Resource Adapter.

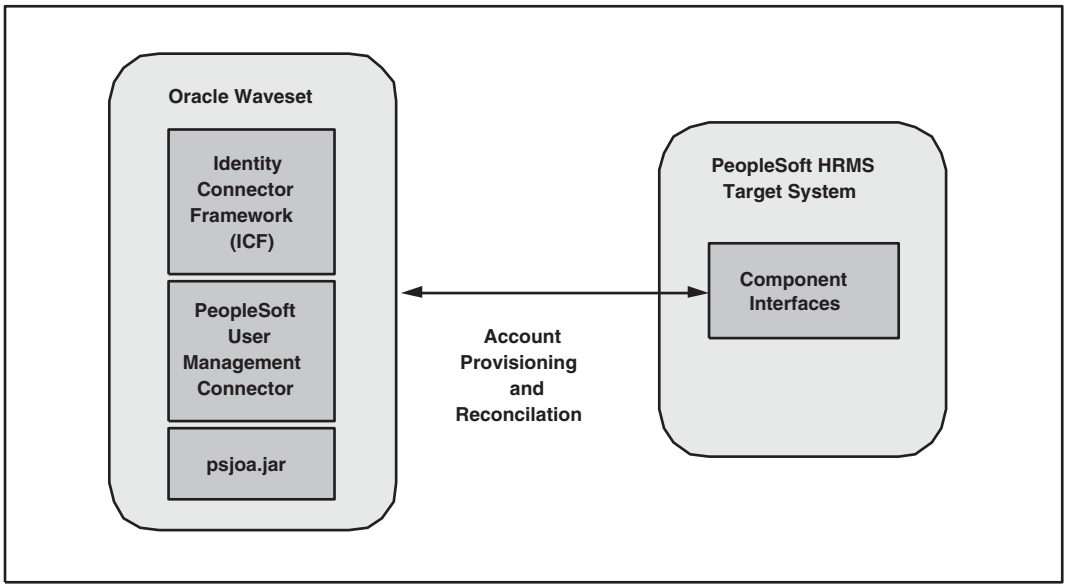
This section also describes:

- PeopleSoft User Management Connector Architecture
- PeopleSoft User Management Connector Features
- PeopleSoft User Management Connector Configuration Properties

8.1.1.1 PeopleSoft User Management Connector Architecture

The following figure shows the PeopleSoft User Management connector architecture.

Figure 8–1 PeopleSoft User Management Connector Architecture



The PeopleSoft User Management connector architecture includes these components:

- Oracle Waveset includes the connector integration files. These files are XML files that provide the configuration information necessary to transform data from a resource to Oracle Waveset. Integration files are sometimes called the connector "glue" code.
- The Identity Connector Framework (ICF) provides basic provisioning, logging, and other functions that Oracle Waveset (and Oracle Identity Manager) connectors can use.
- The PeopleSoft User Management connector uses component interfaces to perform provisioning and reconciliation on the PeopleSoft HRMS target system.

  The connector requires the PeopleSoft Java Object Adapter (`psjoa.jar` file) to access the target system data. The version of the `psjoa.jar` must match the version of the installed PeopleSoft target system.

8.1.1.2 PeopleSoft User Management Connector Features

The PeopleSoft User Management connector supports the operations described in the following table.

Table 8–1 PeopleSoft User Management Connector Operations

Operation	Description
Provisioning	These provisioning operations are supported: ■ Create, update, and delete accounts ■ Enable and disable accounts, if the component interface map defines the enable and disable logic ■ Password update ■ Data loading methods: Import directly from the resource These provisioning operations are **not** supported: ■ Rename account ■ Pass-through authentication ■ Before and after actions
Target reconciliation	Full and incremental reconciliation operations are supported. The component interface must support the `find`, `save` and `get` methods. Reconciliation compares the contents of the account index to what each resource currently contains. Reconciliation can perform these operations: ■ Detect new and deleted accounts. ■ Detect changes in account attribute values. ■ Correlate accounts with Oracle Waveset users. ■ Detect accounts that are not associated with Oracle Waveset users. See also Customizing the Reconciliation Policy for Target Reconciliation.

8.1.1.3 PeopleSoft User Management Connector Configuration Properties

The following table describes the configuration properties for the PeopleSoft User Management connector.

Table 8–2 PeopleSoft User Management Connector Configuration Properties

Property	Description
Host	Hostname or IP address of the PeopleSoft target resource.
TCP Port	Port number on which the PeopleSoft target resource is listening.
User	ID of a PeopleSoft user with the permissions required to invoke methods on the component interfaces.
Password	User's password.
Read/Write component interface key	Component interface that will perform user create, read, and update actions. This component interface must support the `Create`, `Get`, `Save`, and `SetPassword` methods.
Delete user component interface key	Component interface that will perform user deletions. This component interface must support the `Get` and `Save` methods.

8.1.2 Security Considerations for the PeopleSoft User Management Connector

This section provides the following security considerations:

- Supported Connections for the PeopleSoft User Management Connector
- Required Administrative Privileges for the PeopleSoft User Management Connector

8.1.2.1 Supported Connections for the PeopleSoft User Management Connector

Oracle Waveset uses Oracle Jolt to communicate with the PeopleSoft User Management connector.

8.1.2.2 Required Administrative Privileges for the PeopleSoft User Management Connector

The administrative user that connects to the PeopleSoft target system must have privileges to perform any action on the USER_PROFILE and DELETE_USER_PROFILE component interfaces.

If the ROLE_MAINT component interface is configured for the connector, the administrator must have privileges to perform actions on this interface too.

To install the PeopleSoft User Management connector in Oracle Waveset, the user must have Oracle Waveset administrator permissions and access to the file system on the specific application server.

8.1.3 Certified Components for the PeopleSoft User Management Connector

The PeopleSoft User Management connector for Oracle Waveset is certified with the following components:

Table 8–3 Certified Components for the PeopleSoft User Management Connector

Component	Requirement
Oracle Waveset	Oracle Waveset 8.1.1 Patch 6
Target systems	PeopleSoft HRMS 8.9, 9.0, 9.1, and 9.2
PeopleTools	PeopleTools 8.48, 8.49, 8.50, 8.51 and 8.53
Identity Connector Framework (ICF)	ICF 1.0 or later
JDK	JDK 1.5 or later If you are using PeopleTools 8.53, see the note in Section 8.3.2, "Installing the PeopleSoft User Management Connector in the Connector Server," for information related to JDK requirement.

8.1.4 Supported Languages for the PeopleSoft User Management Connector

The PeopleSoft User Management connector is localized in the following languages:

- Arabic
- Chinese (Simplified and Traditional)
- Danish
- French
- German

- Italian
- Japanese
- Korean
- Portuguese (European and Brazilian)
- Spanish

8.2 Migrating to the PeopleSoft User Management Connector from a Resource Adapter

If you currently have the PeopleSoft Component resource adapter deployed, this section describes how to migrate the adapter to the PeopleSoft User Management connector.

The PeopleSoft User Management connector is backwards compatible with the PeopleSoft Component Interface resource adapter. All forms, workflows, and tasks should function the same after you migrate the adapter. However, the connector defines account attributes that were not listed by default in the resource adapter. If you want incorporate any of these attributes into your environment, you might need to update your forms and tasks.

To migrate from a PeopleSoft Component resource adapter, follow these steps:

1. Make sure you have installed the Oracle Waveset patch shown in Certified Components for the PeopleSoft User Management Connector.
2. Log in to the Oracle Waveset Administrator interface.
3. Select the Resources tab and then the Migrate Adapters tab.
4. Follow the Migration Wizard and provide the values for the PeopleSoft User Managementconnector.

   The Migrate Adapters operation automatically migrates the PeopleSoft Component resource adapter to the PeopleSoft User Management connector.

8.3 Deploying the PeopleSoft User Management Connector

You can deploy the PeopleSoft User Management connector either locally in Oracle Waveset or remotely in the Connector Server, as described in the following sections:

- Installing the PeopleSoft User Management Connector in Oracle Waveset
- Installing the PeopleSoft User Management Connector in the Connector Server
- Postinstallation Tasks for the PeopleSoft User Management Connector

8.3.1 Installing the PeopleSoft User Management Connector in Oracle Waveset

To install the PeopleSoft User Management connector in Oracle Waveset, you must have Oracle Waveset administrator privileges and access to the file system on the application server you are using.

To install the PeopleSoft User Management connector in Oracle Waveset, follow these steps:

1. Make sure you have installed the Oracle Waveset patch shown in Certified Components for the PeopleSoft User Management Connector.
2. Stop the Oracle Waveset web application.

3. Explode the `idm.war` file.
4. Copy the PeopleSoft User Management connector bundle JAR file (`org.identityconnectors.peoplesoftintfc-`*`version`*`.jar`) to the `$WSHOME/WEB-INF/bundles` directory of the Oracle Waveset web application.

   In this step, *`version`* represents the connector bundle version. For example: `org.identityconnectors.peoplesoftintfc-1.0.5963.jar`
5. Copy the `peoplesoftintfc-idmglue.jar` file from the PeopleSoft User Management connector `glue code/WEB-INF/lib` directory to the `$WSHOME/WEB-INF/lib` directory.
6. Copy the following XML files from the PeopleSoft User Management connector `glue code/sample/connectors` directory to the `$WSHOME/sample/connectors/peoplesoftintfc-idmglue` directory:
   - `PeopleSoftCompIntfcConnectorUserForm.xml`
   - `postProcess.xml`
   - `preProcess.xml`
   - `resourceWizard.xml`
   - `migration.xml`
7. Copy the `PeopleSoftComponentInterfaces.xml` file from the PeopleSoft User Management connector `glue code/ sample/connectors` directory to the `$WSHome/sample` directory.
8. Copy the `psjoa.jar` file from the PeopleSoft installation media to the `$WSHOME/WEB-INF/lib` directory. The version of the `psjoa.jar` must match the version of the installed PeopleSoft target system.

   Note: You can also copy the `psjoa.jar` file to thePeopleSoft User Management connector bundle `/lib` directory, which will enable support for different versions of the PeopleSoft target system.
9. Create a new `idm.war` file reflecting all of these changes.
10. Deploy the `idm.war` file in the application server you are using.

    For example, start Oracle WebLogic Server. Go to Deployments->Install and provide the path to the new `idm.war` war file. Then click Install.
11. Start the Oracle Waveset web application.
12. Log in to the Oracle Waveset Administrator interface.
13. Click the Configure tab and then Import Exchange File.
14. Select and import the all of the XML files from the previous steps, including `PeopleSoftComponentInterfaces.xml`, `PeopleSoftCompIntfcConnectorUserForm.xml`, `postProcess.xml`, `preProcess.xml`, `resourceWizard.xml`, and `migration.xml`.
15. Edit the Tabbed User Form as described in Configuring the Tabbed User Form.
16. Create a PeopleSoft User Management resource by selecting the PeopleSoft User Management connector and following the Create PeopleSoft User Management Connector Resource wizard.

17. When you configure a resource object, select the following values for attributes:
    - USER_PROFILE_8_4x for `RWCompIntfcKey`
    - `org.identityconnectors.peoplesoft.common.mapping.idm.IDMSAXComponentInterfacesFactory` for Mapping Class Name
    - DELETE_USER_PROFILE for `DelCompIntfcKey`
18. When you set identity system parameters, specify `accountid` for the display name attribute.

After you install the PeopleSoft User Management connector see Configuring Account Attributes for the PeopleSoft User Management Connector.

8.3.2 Installing the PeopleSoft User Management Connector in the Connector Server

The Connector Server requires a JDK to run. For the JDK requirements, see Certified Components for the PeopleSoft User Management Connector. If necessary, set your `JAVA_HOME` environment variable to point to your specific installation.

Note: If you are using PeopleTools 8.53, following is the JDK requirement:

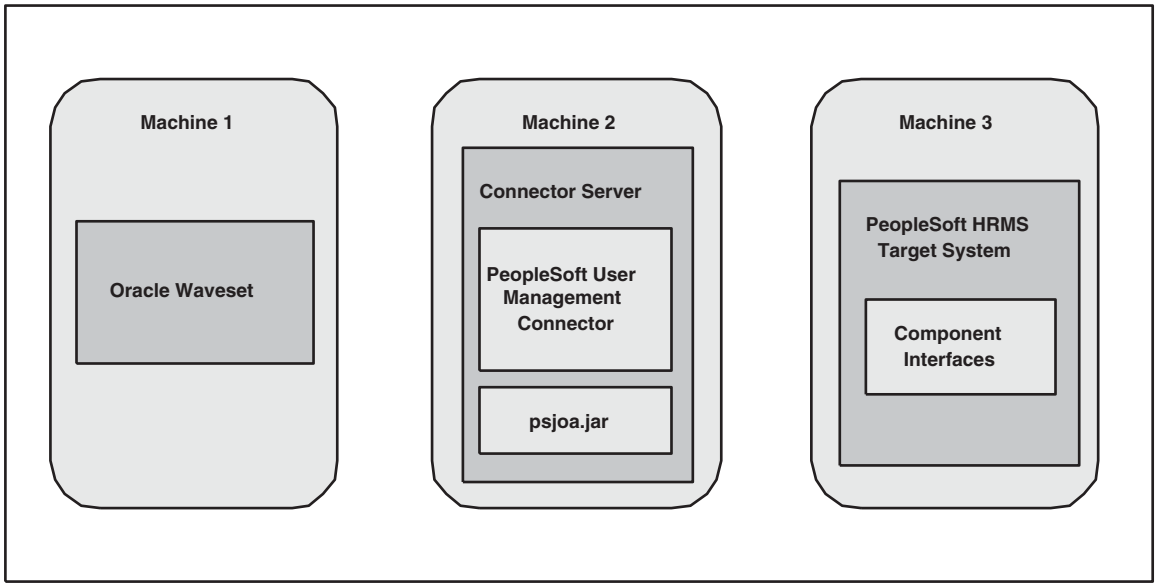
- If you are already using a Connector Server, then it is mandatory to use JDK 1.7.0_02 as the minimum version in the Connector Server.
- If the you are not using Connector Server and Oracle Waveset is not using JDK 1.7.0_02, then follow one of the below mentioned steps:
    - Refer the certification matrix and upgrade the JDK version used by Oracle Waveset to JDK 1.7.0_02 if it is supported.
    - If JDK 1.7.0_02 is not supported for Oracle Waveset, then it is mandatory to use a Connector Server with minimum JDK 1.7.0_02. In addition, create a new Connector Server Definition for the Connector Server and specify it in the PeopleSoft Component Interface Connector Resource.

- PeopleSoft User Management Connector Deployment Architecture With the Connector Server
- Installing and Configuring the Connector Server
- Running the Connector Server on Windows Systems
- Running the Connector Server on UNIX and Linux Systems
- Installing the PeopleSoft User Management Connector in the Connector Server

8.3.2.1 PeopleSoft User Management Connector Deployment Architecture With the Connector Server

If you install the PeopleSoft User Management connector in the Connector Server, the following figure shows the distributed deployment architecture.

Figure 8–2 PeopleSoft User Management Connector Deployment Architecture With the Connector Server



A PeopleSoft User Management connector deployment with the Connector Server includes these components:

- **Machine 1** has Oracle Waveset deployed.
- **Machine 2** has the PeopleSoft User Management connector installed in the Connector Server. The Connector Server is part of the Identity Connector Framework (ICF).

  The PeopleSoft User Management connector bundle is installed in the `CONNECTOR_SERVER_HOME/bundles` directory.

  The appropriate PeopleSoft Java Object Adapter (`psjoa.jar`) file is installed in the `CONNECTOR_SERVER_HOME/lib` directory.
- **Machine 3** has the PeopleSoft HRMS target system deployed.

8.3.2.2 Installing and Configuring the Connector Server

To install and configure the Connector Server, follow these steps:

1. Create a new directory on the machine where you want to install the Connector Server. In this section, `CONNECTOR_SERVER_HOME` represents this directory.
2. Unzip the Connector Server package in your new directory from Step 1. The Connector Server package is available with the Identity Connector Framework (ICF).
3. In the `ConnectorServer.properties` file, set the following properties, as required by your deployment. The `ConnectorServer.properties` file is located in the `conf` directory.

Property	Description
`connectorserver.port`	Port on which the Connector Server listens for requests. The default is 8759.
`connectorserver.bundleDir`	Directory where the connector bundles are deployed. The default is `bundles`.

Property	Description
connectorserver.libDir	Directory in which to place dependent libraries. The default is lib.
connectorserver.usessl	If set to true, the Connector Server uses SSL for secure communication. The default is false. If you specify true, use the following options on the command line when you start the Connector Server: ■ -Djavax.net.ssl.keyStore ■ -Djavax.net.ssl.keyStoreType (optional) ■ -Djavax.net.ssl.keyStorePassword
connectorserver.ifaddress	Bind address. To set this property, uncomment it in the file (if necessary). The bind address can be useful if there are more NICs installed on the machine.
connectorserver.key	Connector Server key.

4. Set the properties in the ConnectorServer.properties file, as follows:
   - To set connectorserver.key, run the Connector Server with the setKey option.
   - For all other properties, edit the ConnectorServer.properties file manually.
5. The conf directory also contains the logging.properties file, which you can edit if required by your deployment.

8.3.2.3 Running the Connector Server on Windows Systems

To run the Connector Server on Windows systems, use the ConnectorServer.bat script as follows:

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

   The ConnectorServer.bat script supports the following options:

Option	Description
/install [*serviceName*] ["-J *java option*"]	Installs the Connector Server as a Windows service. Optionally, you can specify a service name and Java options. If you do not specify a service name, the default name is ConnectorServerJava.
/run ["-J *java option*"]	Runs the Connector Server from the console. Optionally, you can specify Java options. For example, to run the Connector Server with SSL: ConnectorServer.bat /run "-J-Djavax.net.ssl.keyStore=mykeystore.jks" "-J-Djavax.net.ssl.keyStorePassword=*password*"

Option	Description
`/setkey [key]`	Sets the Connector Server key. The `ConnectorServer.bat` script stores the hashed value of the key in the `connectorserver.key` property in the `ConnectorServer.properties` file.
`/uninstall [serviceName]`	Uninstalls the Connector Server. If you do not specify a service name, the script uninstalls the `ConnectorServerJava` service.

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

8.3.2.4 Running the Connector Server on UNIX and Linux Systems

To run the Connector Server on UNIX and Linux systems, use the `connectorserver.sh` script, as follows:

1. Make sure that you have set the properties required by your deployment in the `ConnectorServer.properties` file, as described in Installing and Configuring the Connector Server.
2. Change to the `CONNECTOR_SERVER_HOME/bin` directory.
3. Use the `chmod` command to set the permissions to make the `connectorserver.sh` script executable.
4. Run the `connectorserver.sh` script. The script supports the following options:

Option	Description
`/run [ -Jjava-option ]`	Runs the Connector Server in the console. Optionally, you can specify one or more Java options. For example, to run the Connector Server with SSL: `./connectorserver.sh /run -J-Djavax.net.ssl.keyStore=mykeystore.jks -J-Djavax.net.ssl.keyStorePassword=password`
`/start [ -Jjava-option ]`	Runs the Connector Server in the background. Optionally, you can specify one or more Java options.
`/stop`	Stops the Connector Server, waiting up to 5 seconds for the process to end.
`/stop n`	Stops the Connector Server, waiting up to *n* seconds for the process to end.
`/stop -force`	Stops the Connector Server. Waits up to 5 seconds and then uses the `kill -KILL` command, if the process is still running.
`/stop n -force`	Stops the Connector Server. Waits up to *n* seconds and then uses the `kill -KILL` command, if the process is still running.
`/setKey key`	Sets the Connector Server key. The `connectorserver.sh` script stores the hashed value of *key* in the `connectorserver.key` property in the `ConnectorServer.properties` file.

8.3.2.5 Installing the PeopleSoft User Management Connector in the Connector Server

To install the PeopleSoft User Management connector for Oracle Waveset in the Connector Server, follow these steps:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the PeopleSoft User Management Connector.
2. Stop the Connector Server.
3. Copy the PeopleSoft User Management connector bundle to the `CONNECTOR_SERVER_HOME/bundles` directory.
4. Copy the `psjoa.jar` file to the `CONNECTOR_SERVER_HOME/lib` directory. The version of the `psjoa.jar` must match the version of the installed PeopleSoft target system.

   Note: You can also copy the `psjoa.jar` file to the PeopleSoft User Management connector bundle `/lib` directory, which will enable support for different versions of the PeopleSoft target system.

5. Start the Connector Server.

For information about starting and stopping the Connector Server, see Running the Connector Server on Windows Systems or Running the Connector Server on UNIX and Linux Systems.

8.3.3 Postinstallation Tasks for the PeopleSoft User Management Connector

- Configuring the Tabbed User Form
- Customizing the Reconciliation Policy for Target Reconciliation
- Configuring the PeopleSoft User Management Connector to Support Multiple Versions of the Target System

8.3.3.1 Configuring the Tabbed User Form

To configure the Tabbed User Form after you install the PeopleSoft User Management connector, follow these steps:

1. Go to Oracle Waveset debug page. For example:

   `http://host_name:port/idm/debug`

2. Select User Form from the drop-down box, which is adjacent to List Objects, and then click on List Objects.
3. Search for the Tabbed User Form and then click Edit.

   Note. Do **not** select the Dynamic Tabbed User Form or the Tabbed View User Form.

4. Make the following changes in the Tabbed User Form:

   a. Add the following tag inside the `<Include>` tag:

   ```
   <ObjectRef type='UserForm' name='PeopleSoftCompIntfcConnectorUserForm'/>
   ```

   b. Find `<Field name='_FM_ATTRIBUTES'>` and replace the `<FormRef ...>` attribute with the following attribute:

   ```
   <FormRef name='PeopleSoftCompIntfcConnectorUserForm'>
   <Property name='RESOURCE_NAME' value='name-of-the-resource'/>
   </FormRef>
   ```

8.3.3.2 Customizing the Reconciliation Policy for Target Reconciliation

Before starting target reconciliation, edit the Reconciliation Policy in Oracle Waveset for the following reconciliation action rules:

- **DELETED** — Unlink resource account from user
- **FOUND** — Link resource account to user
- **MISSING** — Unlink resource account from user
- **UNASSIGNED** — Link resource account to user

8.3.3.3 Configuring the PeopleSoft User Management Connector to Support Multiple Versions of the Target System

For some deployments, you might want to configure the PeopleSoft User Management connector for different versions of the PeopleSoft target system. For example, you can configure the connector to perform provisioning operations on both PeopleSoft 8.48 and PeopleSoft 8.49 target systems.

To configure the PeopleSoft User Management connector to support multiple versions of the PeopleSoft target system, follow these steps:

1. Explode the PeopleSoft User Management connector bundle JAR file in a temporary directory. For example:

   ```
   jar -xvf org.identityconnectors.peoplesoftintfc-1.0.5963.jar
   ```

2. In the `META-INF/MANIFEST.MF` file, update the version entry to a new value. For example:

   ```
   ConnectorBundle-Version: 1.0.5964
   ```

3. Copy the appropriate version of the `psjoa.jar` file to the exploded connector bundle `/lib` directory.

   The version of the `psjoa.jar` file must match the specific version of the PeopleSoft target system.

   Note: Make sure that the `psjoa.jar` is **not** in the `$WSHOME/WEB-INF/lib` directory.

4. Create a new connector bundle JAR file with a new version in the file name. For example:

   ```
   jar -cvfm org.identityconnectors.peoplesoftintfc-1.0.5964.jar
   META-INF/MANIFEST.MF .
   ```

   Note. Make sure that the manifest file is not overwritten during this step.

5. Copy the new connector bundle JAR file to the `$WSHOME/WEB-INF/bundles` directory.

   If you are deploying the PeopleSoft User Management connector in the remote Connector Server, copy the new connector bundle JAR file to the `CONNECTOR_SERVER_HOME/bundles` directory.

   Repeat these steps for additional versions of the PeopleSoft target system that you want to support. Each version of the PeopleSoft target system must have a separate connector bundle JAR file.

6. Login to Oracle Waveset and add a resource of type "PeopleSoft component interfaces connector". The new versions will be appear in the resource wizard.

7. Select the specific version and configure the new resource.

To configure the PeopleSoft User Management connector user form for an additional target resource, follow these steps:

1. Open the `PeopleSoftCompIntfcConnectorUserForm.xml` file (which is included with the PeopleSoft User Management connector).
2. Change the form name by replacing "PeopleSoftCompIntfcConnectorUserForm" with a new name such as PeopleSoftAPAC.
3. Update the section head with the new name. For example:

```
<Field>
       <Display class = "SectionHead">
            <Property name = "title" value = "PeopleSoftAPAC"/>
        </Display>
</Field>
```

4. Replace RESOURCE_NAME with RSRC_NAME. This change is required because the parameter is currently shared across forms. A new parameter name must be used for each form reference.
5. Save the form as a new XML file and import the file into Oracle Waveset.

Repeat these steps for each new PeopleSoft target resource that you want to add.

Edit the Tabbed User Form as follows:

1. Go to Oracle Waveset debug page. For example:

   `http://host_name:port/idm/debug`

2. Select User Form from the drop-down box, which is adjacent to List Objects, and then click on List Objects.
3. Search for the Tabbed User Form and then click Edit.
4. Edit the Tabbed User Form as follows:
   1. Add the new entry for the form name (similar to first entry). For example:

```
<Form> <Include>
       <ObjectRef type='UserForm' name='PeoplesoftAPAC'/>
</Include>
...
<Field name='_FM_ATTRIBUTES'>
...
              <FormRef name='PeoplesoftAPAC'>
                <Property name='RSRC_NAME' value='Resource Name'/>
              </FormRef>
... </Field>
```

   2. To display only the form for which the resource is selected, modify the contents associated with the resource. For example:

```
<Field name='_FM_ATTRIBUTES'>
...
         <Field>
           <Disable>
             <not>
               <contains>
                 <ref>accountInfo.assigned</ref>
                 <s>Resource Name</s>
               </contains>
```

```
                </not>
              </Disable>
              <FormRef name='PeoplesoftAPAC'>
                <Property name='RSRC_NAME' value='Resource Name'/>
              </FormRef>
            </Field>
-- </Field>
```

3. To prevent showing additional attributes that are not configured in the User Form, remove the following entry:

   ```
   <FormRef name='MissingFields'/>
   ```

5. Save the Tabbed User Form.

8.4 Using the PeopleSoft User Management Connector

The PeopleSoft User Management connector provides the means to read and write account data on the PeopleSoft resource. The connector must be configured to specify which account attributes Oracle Waveset can manage.

- Account Attributes for the PeopleSoft User Management Connector
- Sample Form for the PeopleSoft User Management Connector
- Connector Component Interfaces for the PeopleSoft User Management

8.4.1 Account Attributes for the PeopleSoft User Management Connector

The account attributes for the PeopleSoft User Management connector resource depend on the component interface being managed.

Each entry of the schema map should have a Resource User Attribute name that matches one of the entries in the "properties" list defined for the component interface in the Component Interface Map. When editing the schema map, you can click the **Test Configuration** button to verify an appropriate match can be found.

If the Resource User Attribute name matches a collection property in the component interface map, the value for the account attribute will be a GenericObject (EmbeddedObject) representation of the collection. For example for manipulating collection properties, see the sample `accounts[$(RESOURCE_NAME)].Roles`, `accounts[$(RESOURCE_NAME)].IDTypes`, and `accounts[$(RESOURCE_NAME)].EmailAddresses` fields in the `PeopleSoftCompIntfcConnectorUserForm`.

> **Note:** The default schema map entries that are defined for a new resource instance are appropriate only when used with the default USER_PROFILE and DELETE_USER_PROFILE component interface maps. If you change these maps, or create your own, then you must change your schema map accordingly.

All account attributes are of type String unless otherwise stated.

The account attributes are described in Table 8–4.

Table 8–4 Account Attributes for the PeopleSoft User Management Connector

Oracle Waveset User Attribute	Resource User Attribute	Description
User Description	UserDescription	Description of the user.
User Symbolic ID	SymbolicID	User's symbolic ID.
Language Code	LanguageCode	Language code associated with this user.
UserIDAlias	UserIDAlias	Another ID for the user.
AlternateUserID	AlternateUserID	ID that sends workflow items.
AccountLocked	AccountLocked	Integer. Sets the status of the account. A value of 1 indicates the account is locked (disabled). A value of 0 indicates the account is not locked (enabled).
EffectiveDateFrom	EffectiveDateFrom	Start date associated with the AlternateUserID attribute.
EffectiveDateTo	EffectiveDateTo	End date associated with the AlternateUserID attribute.
Currency Code	CurrencyCode	Type of currency for this user's monetary transactions.
MultiLanguageEnabled	MultiLanguageEnabled	Optional attribute that indicates if the user is set up to use PeopleSoft with multiple languages.
NavigatorHomePermissionList	NavigatorHomePermissionList	Integer. Type of permission list that navigates to a specified home page.
PrimaryPermissionList	PrimaryPermissionList	Integer. Primary Permissions that are assigned to the user.
ProcessProfilePermissionList	ProcessProfilePermissionList	Integer. Type of permission list that specifies the process-level security for the user.
RowSecurityPermissionList	RowSecurityPermissionList	Integer. Provides the row-level security.
Employee ID	Employee ID	Employee ID assigned to the user. A user can be assigned one or more of these ID types: EmplID, Customer ID, Customer SetID, Vendor ID, and Vendor SetID. For example, the same user be assigned an EmplID and a Customer ID. Different versions of PeopleSoft have different values for the Employee ID field name: ■ PeopleSoft 8.50 and 8.51: EMPL ID ■ PeopleSoft 8.48 and 8.49: EMPLID See Modifying the User Form for Different Versions of PeopleSoft.
Customer ID	Customer ID	Customer ID assigned to the user. The ID type is determined by the IDType attribute, which is a complex attribute determined by IDType=CST followed by AttributeName set to Customer ID.

Table 8–4 (Cont.) Account Attributes for the PeopleSoft User Management Connector

Oracle Waveset User Attribute	Resource User Attribute	Description
`Customer SetID`	`Customer SetID`	Customer Set ID assigned to the user. Different versions of PeopleSoft have different values for the Customer Set ID field name: ■ PeopleSoft 8.50 and 8.51: `Set ID` ■ PeopleSoft 8.48 and 8.49: `SetID` See Modifying the User Form for Different Versions of PeopleSoft.
`Vendor ID`	`Vendor ID`	Vendor ID assigned to the user. The ID type is determined by the `IDType` attribute, which is a complex attribute determined by `IDType=VND` followed by `AttributeName` set to `Vendor ID`.
`Vendor SetID`	`Vendor SetID`	Vendor Set ID assigned to the user. Different versions of PeopleSoft have different values for the Vendor Set ID field name: ■ PeopleSoft 8.50 and 8.51: `Set ID` ■ PeopleSoft 8.48 and 8.49: `SetID` See Modifying the User Form for Different Versions of PeopleSoft.
`Roles`	`Roles`	List of roles assigned to the user.
`EmailUser`	`EmailUser`	Complex email attribute name that contains space for all five email types and their primary markers.
`Business Email Address`	Complex attribute determined by `EmailType==BUS`	User's business email address. If this address is the primary email address, check `Is Business Primary Email Address Type`.
`Work Email Address`	Complex attribute determined by `EmailType==WORK`	User's work email address. If this address is the primary email address, check `Is Work Primary Email Address Type`.
`HOME Email Address`	Complex attribute determined by `EmailType==HOME`	User's home email address. If this address is the primary email address, check `Is Home Primary Email Address Type`.
`BlackBerry Email Address`	Complex attribute determined by `EmailType==BB`	User's BlackBerry email address. If this address is the primary email address, check `Is BlackBerry Primary Email Address Type`.
`Other Email Address`	Complex attribute determined by `EmailType==OTH`	Any other email address of the user. If this address is the primary email address, check `Is Other Primary Email Address Type`.

Table 8–4 (Cont.) Account Attributes for the PeopleSoft User Management Connector

Oracle Waveset User Attribute	Resource User Attribute	Description
SupervisingUserID	SupervisingUserID	User ID at which workflow items are to be redirected.
ReassignUserID	ReassignUserID	User ID to reassign work to.
ReassignWork	ReassignWork	Integer. Indicates whether work for this user should be reassigned. The value 1 enables this feature.

Note: Configuring Email Addresses. If any of the email address attributes are provided for a user, you must specify one (but only one) as the primary email address by checking the appropriate box.

8.4.1.1 Configuring Account Attributes for the PeopleSoft User Management Connector

After you install the PeopleSoft User Management connector, configure the account attributes as follows:

1. Edit the PeopleSoft User Management connector user form:
    1. Add the attributes to be displayed in the user form in the `PeopleSoftCompIntfcConnectorUserForm.xml` file.

       This file is available in the following directory:

       `$WSHOME/sample/connectors/peoplesoftintfcidmglue/`

    2. Copy the PeopleSoft User Management connector bundle to the `$WSHOME/WEB-INF/bundles` directory.
    3. Log in to the Oracle Waveset Administrator interface.
    4. Click the Configure tab.
    5. Click Import Exchange File and then import the `PeopleSoftCompIntfcConnectorUserForm.xml` file.
2. Link the attributes to the PeopleSoft target system.
    1. Add the newly added attributes in the PeopleSoft User Management connector user form to the `$WSHOME/sample/PeopleSoftComponentInterfaces.xml` file.
    2. In the Oracle Waveset web application, click the Configure tab.
    3. Click Import Exchange File and then import the `PeopleSoftComponentInterfaces.xml` file.
    4. Add values for the new attributes. For example, if new attributes are added for the USER_PROFILE_8_4x object, provide a value for the `RWIntfcKey` attribute USER_PROFILE_8_4x attributes.
3. Map the entities in Oracle Waveset and the PeopleSoft target system:
    1. In the Oracle Waveset web application, click the Resources tab and then the name of the PeopleSoft User Management connector resource object. The resource type is PeopleSoft Component Interface Connector.
    2. Click Next.

The Read/Write component interface key USER_PROFILE_8_4x is selected by default. The component interface metadata for the profile is displayed in the text area.

3. Add the newly added attributes and then map each Identity System User Attribute to the corresponding Resource User Attribute.
4. When you are finished mapping the attributes, click Next and then Save.

8.4.2 Sample Form for the PeopleSoft User Management Connector

The `PeopleSoftCompIntfcUserForm.xml` sample form is provided in the `$WSHOME/sample/connectors/peopleoftintfc-idmglue` directory.

The `PeopleSoftCompIntfcUserForm.xml` is configured by default for the USER_PROFILE component interface. You can edit or replicate this form as required for your deployment.

8.4.2.1 Modifying the User Form for Different Versions of PeopleSoft

The following attributes have different form field names for different versions of the PeopleSoft target system:

Attribute	PeopleSoft Version and Form Field Name Value
Employee ID	■ PeopleSoft 8.50 and 8.51: `EMPL ID` ■ PeopleSoft 8.48 and 8.49: `EMPLID`
Customer Set ID and Vendor Set ID	■ PeopleSoft 8.50 and 8.51: `Set ID` ■ PeopleSoft 8.48 and 8.49: `SetID`

The `PeopleSoftCompIntfcUserForm.xml` sample form has definitions for PeopleTools 8.50 and 8.51. If you are using PeopleSoft 8.48 or 8.49, follow these steps to modify the definitions:

1. Edit the `PeopleSoftCompIntfcUserForm.xml` file and change the form field name values for the Employee ID, Vendor Set ID, and Customer Set ID attributes, as required by your deployment.

   For example, the following change is for the Employee ID attribute:

```
                <Field name =
"accounts[$(RESOURCE_NAME)].IDTypes_raw[IDType==EMP].Attributes.AttributeName">
                    <Default>
                        <s>EmplID</s>
                         <!-- Changed the name from Empl ID to EmplID -->
                    </Default>
                </Field>
                <Field name =
"accounts[$(RESOURCE_NAME)].IDTypes_raw[IDType==EMP].Attributes.AttributeValue"
>
                    <Display class = "Text">
                        <Property name = "title" value = "Employee ID"/>
                    </Display>
                </Field>
```

2. In Oracle Waveset, import the modified `PeopleSoftCompIntfcUserForm.xml` file.

8.4.3 Connector Component Interfaces for the PeopleSoft User Management

The PeopleSoft User Management connector performs user provisioning by invoking methods and setting properties on PeopleSoft component interfaces. Component interface definitions are assigned in the PeopleSoft Component Interface configuration object. This object can be modified through the debug pages or with the Waveset IDE. You can also edit a copy of the `$WSHOME/sample/PeopleSoftComponentInterfaces.xml` file and load that file into Oracle Waveset.

This sectgion includes the following information about configuring and implementing component interfaces with the PeopleSoft User Management connector:

- Creating Component Interface Map Definitions
- Customizing PeopleSoft Component Interface Resource Objects
- Adding FIND Method Support to the USER_PROFILE Component Interface
- Adding New ID Types for the USER_PROFILE Component Interface
- Adding New Attributes for the USER_PROFILE Component Interface

8.4.3.1 Creating Component Interface Map Definitions

The component interface map contains the list of component interfaces available to the connector. The `interfaces` object contains a list of component interfaces. If you have a custom component interface, you must define your own component interface definition in the map. Edit the PeopleSoft Component Interfaces Configuration object and add your definition as an additional Object into the `<List>` element under the `<Attribute name='interfaces'>` element.

Each available component interface has its own definition. Key elements of a component interface definition include:

- `name`. The label of a component interface. It often matches the value of the `componentInterface` attribute, but this is not a requirement. The value will be displayed in the drop-down menu on the connector's Resource Parameters page.
- `componentInterface` attribute. The name of the component interface, as defined in PeopleSoft.
- `getKey` attribute. The name of the component interface property that is set when performing a PeopleSoft GET operation. If `getKey` is not defined, then the `key` attribute is used instead.
- `findKey` attribute. The name of the component interface property that is set when performing a PeopleSoft FIND operation. If `findKey` is not defined, then the `key` attribute is used instead.
- `createKey` attribute. The name of the component interface property that is set when performing a PeopleSoft CREATE operation. If `createKey` is not defined, then `key` attribute is used instead.
- `key` attribute. Deprecated. Use `getKey`, `findKey`, or `createKey` instead.
- `properties` attribute. A list of properties that can be read or set from the PeopleSoft component interface.

  Each Object in the *properties* list must have the following attribute:

  - `name`. The name of the property. This must match exactly with the name of a property exposed by the PeopleSoft component interface identified by the

componentInterface property. The names of the properties are candidates to be listed as resource user attributes on the Account Attributes page.

If this a collection property, then you must define additional attributes. A collection property defines its key property and its own nested set of simple and/or complex properties:

- isCollection attribute. If the property is a collection, then set this to true.
- key attribute. If the property is a collection, set this to the name of the property that uniquely identifies each item of the collection.
- properties attribute. The list of properties that can be read/set for each item of the collection. To support arbitrary complexity, each member of this list is an Object with the same allowed attributes as the parent. That is, it can contain its own name, isCollection, key, and properties attributes.

disableRule attribute. An Object that defines the logic to compute and set the user disable state. This attribute contains the following attributes

- property attribute. The property to check. The value must be listed in the properties attribute for the componentInterface object.
- trueValue attribute. A value that indicates the user is disabled.
- falseValue attribute. A value that indicates the user is enabled.

supportedObjectTypes attribute. A list of Oracle Waveset resource objects types that can be accessed through the connector. Each object defines a set of features.

- features attribute. A list supported features. Possible feature types include view, get, list, find, create, saveas, update, rename, and delete.

8.4.3.1.1 Default Component Interfaces Supported The default Component Interface configuration object defines the following interfaces:

- USER_PROFILE. Performs create, read, and update actions.
- DELETE_USER_PROFILE. Deletes user accounts.
- ROLE_MAINT. Adds support for PeopleSoft roles.

USER_PROFILE Component Interface

The default USER_PROFLE component interface definition is used to perform create, read, and update actions. The key and findKey attributes are set to UserID, because the USER_PROFILE component interface assigns the UserID field for the GETKEYS and FINDKEYS keys.

The default definition for the USER_PROFILE component interface does not define all of the possible properties. It has been simplified to include those used in the sample user form. If you need to add more resource user attributes to the Account Attributes page, then the component interface definition must be updated first. A resource user attribute cannot be added to that page unless it is listed in the component interface definition.

Most properties are defined in USER_PROFILE are simple objects. However, the IDTypes and Roles objects are collections and can have multiple values. IDTypes contains a collection of its own, Attributes. These objects must include the isCollection attribute, the key name for the collection, and at least one property.

DELETE_USER_PROFILE Component Interface

The DELETE_USER_PROFILE component interface definition is used to delete user profile definitions. The OPRID key determines which user profile is to be deleted. Since the component interface does not have properties, none are listed in the definition.

ROLE_MAINT Component Interface

The ROLE_MAINT component interface definition is part of a sample implementation that illustrates how Oracle Waveset can be configured to list role resource objects. Other resource objects can be listed by following the general guidelines listed below and modifying the ROLE_MAINT example to match your requirements.

The ROLE_MAINT component interface definition has the following characteristics:

- The `findKey` and `getKey` attributes are assigned to ROLENAME because ROLENAME is the primary key for FINDKEYS and GETKEYS.
- DESCR and ROLESTATUS are also keys in FINDKEYS, but since they are not primary keys, they are not listed as values for `findKey`. Instead, they are listed in the `properties` section.
- The `supportedObjectTypes` attribute defines the Role object. The Role object supports the find and get features.

8.4.3.2 Customizing PeopleSoft Component Interface Resource Objects

The PeopleSoft Component Interface resource XML can be edited so that resource objects can be managed. Use the debug pages or the Waveset IDE to add an `ObjectType` element. For example, to add support for the Role resource object, add an `ObjectType` element similar to the following example:

```
<ObjectTypes>
<ObjectType name='Role' icon='role'>
   <ObjectFeatures>
      <ObjectFeature name='find'/>
   </ObjectFeatures>
   <ObjectAttributes idAttr='ROLENAME' displayNameAttr='ROLENAME'
descriptionAttr='DESCR'>
      <ObjectAttribute name='ROLENAME' type='string'/>
      <ObjectAttribute name='DESCR' type='string'/>
      <ObjectAttribute name='ROLESTATUS' type='string'/>
   </ObjectAttributes>
</ObjectType>
</ObjectTypes>
```

The `ObjectType` `name` (for example, Role) must match the name of one of the objects in the `supportedObjectTypes` list of exactly one component interface definition. Each `ObjectFeature` (for example, find) must have a corresponding feature in the `features` list in that same `supportedObjectTypes`. The matched component interface is used to perform the resource feature. (If there are multiple matches, the first one found will be used.)

The following example is part of the component interface definition for the ROLE_MAINT component interface in the component interface map. Note that the Object name Role is found and that an item in the features list is named find.

```
<Attribute name='supportedObjectTypes' >
   <List>
      <Object name='Role'>
         <Attribute name='features' >
            <List>
               <Object name='find' />
```

```
            <Object name='get' />
          </List>
        </Attribute>
      </Object>
   </List>
</Attribute>
```

User Form

The following user form fragment can be used to retrieve a list of PeopleSoft roles. Note that ROLENAME and DESCR attributes are being fetched.

```
<invoke name='getResourceObjects' class='com.waveset.ui.FormUtil'>
   <ref>:display.session</ref>
   <s>Role</s>
   <s>PeopleSoft Component Interface</s>
   <map>
      <s>searchAttrsToGet</s>
      <list>
         <s>ROLENAME</s>
         <s>DESCR</s>
      </list>
    </map>
</invoke>
```

8.4.3.3 Adding FIND Method Support to the USER_PROFILE Component Interface

The default USER_PROFILE component interface does not support the FIND method. However, the PeopleSoft Component Interface adapter requires the FIND method in order to support account iteration and list.

To add FIND method support to an existing USER_PROFILE component interface, follow these steps:

1. Load the USER_PROFILE component interface in the PeopleSoft Application Designer.
2. On the left window (which shows the USERMAINT Component), select the OPRID field under the PSOPRDEFN_SRCH object.

   Drag this field over to the right window (which shows the USER_PROFILE component interface).

   When you drop the field, a new key called FINDKEYS will be created in the USER_PROFILE CI. Under that key, there will be a sub-key called OPRID.
3. Right-click on the OPRID name under FINDKEYS, and select **Edit Name**. Change the name to UserID.
4. Right click on USER_PROFILE CI and select **Component Interface Properties**. Select the **Standard Methods** tab, then select the **Find** checkbox. Click OK to close the **Component Interface Properties** dialog.
5. Save your changes to the USER_PROFILE component interface.

   The Find method is now visible under the METHODS field for the component interface. To verify the functionality of the new FIND method, right-click on the component interface and select **Test Component Interface**.

Note: A PeopleSoft administrator should grant Full Access to the FIND method for the component interface (in addition to the Create, Get, Save, and SetPassword methods).

8.4.3.4 Adding New ID Types for the USER_PROFILE Component Interface

To add a new ID type for the USER_PROFILE component interface, follow these steps:

1. In the PeopleSoft Application Designer, use the component interface tester to determine the values for ID Type and Attribute Name.

   The definition for ID Types is already in the `PeopleSoftComponentInterfaces.xml` file, so you don't have to modify this file.

2. Modify the `PeopleSoftCompIntfcConnectorUserForm.xml` file to include the new ID type.

   Since the ID Type is a complex attribute, you must define a field for the ID type in the user form and then add a conditional expression. The following example shows how a new field is added for ID type "Equation Sql Auth Classes" found in the ID tab of the use profile followed by a conditional expression:

```
<Field name =
"accounts[$(RESOURCE_NAME)].IDTypes_raw[IDType==EQS].Attributes.AttributeName">
    <Default>
        <s>Operator Alias Value</s>
    </Default>
</Field>
<Field name =
"accounts[$(RESOURCE_NAME)].IDTypes_raw[IDType==EQS].Attributes.AttributeValue"
>
    <Display class = "Text">
        <Property name = "title" value = "EQS ID"/>
    </Display>
</Field>

<cond>
   <notnull>
       <ref>
           accounts[
           <ref>RESOURCE_NAME</ref>
           ].IDTypes_raw[IDType==EQS].Attributes.AttributeValue
       </ref>
   </notnull>
   <new class = "com.waveset.object.GenericAttribute">
       <block>
           <defvar name = "IDTypes">
               <new class = "com.waveset.object.GenericObject"/>
           </defvar>
           <set>
               <ref>IDTypes</ref>
               <s>Attributes</s>
               <block>
                   <defvar name = "IDTypes_Attribute">
                       <new class = "com.waveset.object.GenericObject"/>
                   </defvar>
                   <set>
                       <ref>IDTypes_Attribute</ref>
                       <s>AttributeValue</s>
```

```
                        <ref>
                            accounts[
                            <ref>RESOURCE_NAME</ref>

].IDTypes_raw[IDType==EQS].Attributes.AttributeValue
                        </ref>
                    </set>
                    <set>
                        <ref>IDTypes_Attribute</ref>
                        <s>AttributeName</s>
                        <ref>
                            accounts[
                            <ref>RESOURCE_NAME</ref>

].IDTypes_raw[IDType==EQS].Attributes.AttributeName
                        </ref>
                    </set>
                    <ref>IDTypes_Attribute</ref>
                </block>
            </set>
            <set>
                <ref>IDTypes</ref>
                <s>IDType</s>
                <ref>
                    accounts[
                    <ref>RESOURCE_NAME</ref>
                    ].IDTypes_raw[IDType==EQS].IDType
                </ref>
            </set>
            <ref>IDTypes</ref>
        </block>
    </new>
 </cond>
```

After you are finished, save the `PeopleSoftCompIntfcConnectorUserForm.xml` file.

3. Login to Oracle Waveset, navigate to Configure -> Import Exchange File, and then import the modified `PeopleSoftCompIntfcConnectorUserForm.xml` file.

8.4.3.5 Adding New Attributes for the USER_PROFILE Component Interface

To add a new attribute for the USER_PROFILE component interface, follow these steps:

1. Using the PeopleSoft Application Designer, add the new attribute to the PROPERTIES list and save it. Note the value of the new attribute under the Name column, because this value will be specified later in the XML file.
2. Using the Component Interface tester, ensure that you can perform find, create, get, save operations for the newly added attribute.
3. Define the new attribute in the `PeopleSoftComponentInterfaces.xml` file. Configurations added in the XML file for "Default Mobile page" are present in the user profile General tab. For example:

```
 <Attribute name="properties">
     <List>
...
<Object name="MPDEFAULMP"/>
...
```

```
    </List>
  </Attribute>
```

After you are finished, save the `PeopleSoftComponentInterfaces.xml` file.

4. Login to Oracle Waveset, navigate to Configure -> Import Exchange File, and then import the modified `PeopleSoftComponentInterfaces.xml` file.
5. Modify the `PeopleSoftCompIntfcConnectorUserForm.xml` file. Configurations added in the XML file for the "Default Mobile page" field are present in the user profile General tab. For example:

```
...
<Field name = "accounts[$(RESOURCE_NAME)].MPDEFAULMP">
    <Display class = "Text">
       <Property name = "title" value = "Default Mobile page"/>
    </Display>
</Field>
...
```

After you are finished, save the `PeopleSoftCompIntfcConnectorUserForm.xml` file.

6. Login to Oracle Waveset, navigate to Configure -> Import Exchange File, and then import the modified `PeopleSoftCompIntfcConnectorUserForm.xml` file.

8.5 Troubleshooting the PeopleSoft User Management Connector

This section describes the following troubleshooting options:

- Section 8.5.1, "Setting Oracle Waveset Trace Options"
- Section 8.5.2, "Increasing the Resource Timeout for Target Reconciliation"

8.5.1 Setting Oracle Waveset Trace Options

Use the Oracle Waveset debug pages to set trace options on the following class:

`org.identityconnectors.peoplesoft.compintfc`

To set the logging options for the Connector Server, edit the properties in the `CONNECTOR_SERVER_HOME/conf/logging.properties` file. See Installing and Configuring the Connector Server.

For more information about enabling Oracle Waveset tracing and setting the tracing levels, see "Tracing Waveset Objects and Activities," in the *Oracle Waveset 8.1.1 System Administrator's Guide* in the following library:

`http://docs.oracle.com/cd/E19225-01/index.html`

8.5.2 Increasing the Resource Timeout for Target Reconciliation

To increase the Resource Timeout value, which is needed for target reconciliation, follow these steps:

1. Go to the Oracle Waveset debug page. For example:

   `http://host:port/idm/debug`

2. Change to configuration against the List Objects option.
3. Click on List Objects.

4. Click Edit on Reconcile Configuration.
5. Change List Timeout by adding 40 seconds.
6. Save your change and then run reconciliation.

8.6 Known Issues for the PeopleSoft User Management Connector

The PeopleSoft User Management connector has the following known issue:

8.6.1 Bug 13348197: PeopleSoft User Management Connector Reconciliation Doesn't Include Names Starting With Wildcard Characters

During the reconciliation of user profiles, users with names starting with a wildcard character such as _ or `%` do not get reconciled in Oracle Waveset. For example, a name such as `_firstname_lastname` does not get reconciled. This problem occurs because the escape characters specified in PeopleBooks such as `//`, `-`, `"`, and _ are not functioning properly.

Workaround. Do not specify a user name that starts with a wildcard character that must be escaped during reconciliation.

9

Oracle Waveset Connector for SAP User Management

This chapter includes the following information about the SAP User Management connector for Oracle Waveset:

- About the SAP User Management Connector
- Deploying the SAP User Management Connector
- Using the SAP User Management Connector
- Troubleshooting the SAP User Management Connector
- Known Issues for the SAP User Management Connector

9.1 About the SAP User Management Connector

- Overview of the SAP User Management Connector
- Security Considerations for the SAP User Management Connector
- Certified Components for the SAP User Management Connector
- Supported Languages for the SAP User Management Connector

9.1.1 Overview of the SAP User Management Connector

The SAP User Management connector provides provisioning and reconciliation for SAP target systems. For the supported SAP target systems, see Certified Components for the SAP User Management Connector.

The SAP User Management connector uses the Business Application Programming Interface (BAPI) to send requests to the SAP target system. The BAPI is available in the SAP Java Connector (JCo) package.

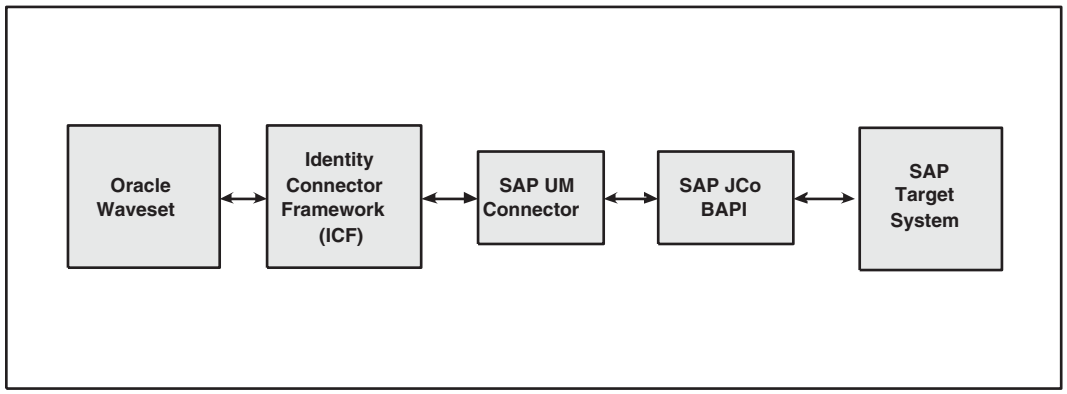
This section provides the following additional information about the SAP User Management connector:

- SAP User Management Connector Architecture
- SAP User Management Connector Features
- SAP User Management Connector Resource Configuration Parameters

9.1.1.1 SAP User Management Connector Architecture

The following figure shows the SAP User Management connector architecture.

Figure 9–1 SAP User Management Connector Architecture



The SAP User Management connector architecture includes these components:

- Oracle Waveset includes the connector integration files. These files are XML files that provide the configuration information necessary to transform data from a resource to Oracle Waveset. Integration files are sometimes called the connector "glue" code.
- The Identity Connector Framework (ICF) provides basic provisioning, logging, and other functions that Oracle Waveset (and Oracle Identity Manager) connectors can use.
- The SAP User Management connector uses the Business Application Programming Interface (BAPI) to send requests to the SAP target system. The BAPI is available in the SAP Java Connector (JCo) package.

If you are deploying the SAP User Management connector in the Connector Server, see also SAP User Management Connector Deployment Architecture With the Connector Server.

9.1.1.2 SAP User Management Connector Features

The SAP User Management connector supports the operations described in the following table.

Table 9–1 SAP User Management Connector Operations

Operation	Description
Provisioning	Operations include: ▪ Create, update, and delete users ▪ Enable and disable users ▪ Change and reset password
Reconciliation	Operations include: ▪ Target full reconciliation ▪ Target incremental reconciliation ▪ Search for accounts

9.1.1.2.1 Support for Failover of SAP Target Systems If Logon Group is configured in the SAP target system, both failover and load balancing are supported. The SAP Message Server provides this support.

If Logon Group is configured for the SAP system, information about the Message Server is provided in the Resource.

For more information about the Message Server, see the SAP documentation for your specific SAP target system.

9.1.1.3 SAP User Management Connector Resource Configuration Parameters

The SAP User Management connector resource configuration parameters are described in the following sections:

- SAP Administrator Credentials Parameters
- SAP Secure Network Communications (SNC) Parameters
- SAP Destination Connection Tuning Parameters
- SAP Central User Administration (CUA) Parameters
- Miscellaneous Optional Parameters

In the following tables, required parameters are noted in the description. Other attributes are optional.

9.1.1.3.1 SAP Administrator Credentials Parameters The following table describes the SAP administrator credential parameters used by the SAP User Management connector.

Table 9–2 SAP Administrator Credentials Parameters

Parameter	Type	Description
SAP Destination Name	String	Unique resource name that defines the destination to be created. Required.
Host	String	Host name of the resource. Required.
System Number	String	SAP System Number. Required.
SAP Client	String	SAP Client setting. Default is 000. Required.
User	String	When using normal authentication, a user name that has permissions to create new accounts. Required for normal authentication.
Password	String	When using normal authentication, password of the User account. Required for normal authentication.
Language	String	Server language setting. Default is EN (English).

9.1.1.3.2 SAP Secure Network Communications (SNC) Parameters The following table describes the SAP SNC parameters. Use these parameters to enable and configure SNC for secure communication between Oracle Waveset and the SAP target system.

See also Configuring Secure Network Communications (SNC) for the SAP User Management Connector.

Table 9–3 SAP Secure Network Communications (SNC) Parameters

Parameter	Type	Description
Enable SAP SNC	Boolean	Enable secure communication using SNC between Oracle Waveset and the SAP target system instead of regular authentication.
SNC Library Path	String	When using SNC, the full path to the SNC cryptographic library file including the file extension (.so, .a, or .dll).
SNC Partner Name	String	When using SNC, the name of the SAP system that is known to the SNC environment. This string value looks like a DN but is prepended with `p:`. For example: `p:CN=SAPHost, OU=IDM, O=Example, C=US`
SNC Protection Level	String	When using SNC, the level of privacy for this connection: ■ 1 is minimum. ■ 9 is maximum. Both sides of the connection must specify the same level of protection.
SNC Name	String	When using SNC, name for the client that is known to the SNC environment. This string value looks like a DN but is prepended with `p:`. For example: `p:CN=Waveset, OU=IDM, O=Example, C=US`
SNC X500 Certificate	String	When using SNC, the X509 certificate. You must delete the BEGIN CERTIFICATE and END CERTIFICATE lines and remove all newline characters from the certificate.

9.1.1.3.3 SAP Destination Connection Tuning Parameters The following table describes the SAP destination connection tuning parameters used by the SAP User Management connector.

Table 9–4 SAP Destination Connection Tuning Parameters

Parameter	Type	Description
Configure Connection Tuning	Boolean	Allows the connection properties to be customized when the SAP Destination is configured.
Max Active Connections	Integer	Maximum number of active connections that can be simultaneously created for a destination.
Pool Capacity	Integer	Maximum number of idle connections that can be kept open by the destination.
Connection Expire Time	Integer	Freed connections held by the destination that can be closed after this amount of time. Specified in milliseconds.
Check Released Connections Period	Integer	Released connections are checked for expiration after waiting for this time period. Specified in milliseconds.
Max Connection Wait Time	Integer	Maximum time to wait for a connection. Specified in milliseconds.
JCO Trace Level	Integer	Level of SAP JCo tracing to enable. Enter 0 or any positive integer up to and including 10.
JCO Trace Directory	String	Absolute path to the directory where the trace files will be created.

9.1.1.3.4 SAP Central User Administration (CUA) Parameters The following table describes the CUA parameters used by the SAP User Management connector.

Table 9–5 SAP Central User Administration (CUA) Parameters

Parameter	Type	Description
Enable CUA	Boolean	If set to true, the connector can manage subsytems as well as roles and profiles on those subsystems.
CUA Child Password Check Delay	Integer	Specifies the milliseconds to delay before checking the propagation of the initial password change to the child systems. This parameter is not used unless CUA is enabled. Default is 1000 milliseconds.
CUA Child Initial Password Change Function Module	String	Name of the Remote Enabled function module that changes the initial password for a user on all CUA child systems. This parameter is not used unless CUA is enabled. If the value is not set, password changes will only applied to the CUA system. Setting productive passwords on CUA child systems will also automatically fail without this setting.
CUA Child Password Change Function Module	String	Name of the Remote Enabled function module that changes the productive password for a user on a CUA child system. This parameter is not used unless CUA is enabled.
CUA Child Password Check Function Module	String	Name of the Remote Enabled function module that checks the setting of the initial password on a child system. This parameter is not used unless CUA is enabled. **Note**: If the value is not set, no checks will be performed, which could cause failures when setting a productive password.

9.1.1.3.5 SAP Password Change Parameters The following table describes the SAP password change parameters used by the SAP User Management connector.

Table 9–6 SAP Password Change Parameters

Parameter	Type	Description
Return SAP Temporary Passwords on Failure	Boolean	Flag that determines whether the plain text temporary password is returned when an error occurs during user password changes. This flag is necessary because two separate password changes are required for a user password change. The first change is done as an admin with a temporary password. The second change is done with the new password to prevent the password from being expired. If the system fails after the first change but before the second change, the connector returns the temporary password in plain text so that the user can see their current password.
Upper Case Passwords	Boolean	When selected, converts the password to uppercase format before sending to the resource.
Use SAP Temporary Passwords	Boolean	Allow SAP to generate a password for use in setting a user's password as expired or unexpired. This option requires installation of SAP Note 832661.

9.1.1.3.6 Miscellaneous Optional Parameters The following table describes miscellaneous optional parameters used by the SAP User Management connector.

Table 9–7 Miscellaneous Optional Parameters

Parameter	Type	Description
Filtered Accounts	String	Listed accounts cannot be edited, created, or listed.
SAP Retry Count	Integer	Number of times to retry a failed operation. A failure could occur due to a network outage or some other anomaly. Default is 5.
SAP Retry Wait Time	Integer	Number of milliseconds to wait before attempting a new operation. Default is 1000 milliseconds.
Temporary Password	String	Temporary password to use while doing password changes.
User Provides Password On Change	Boolean	If selected, the user is required to specify their current SAP password when changing the password.
Eat Non Update Create	Boolean	If set to true, the connector does not throw an exception when it sends an parameter that cannot be created or updated. Instead, the connector just processes the remaining parameters. If set to false, the connector throws an exception without processing.

9.1.2 Security Considerations for the SAP User Management Connector

This section provides the following security considerations for the SAP User Management connector:

- Secure Communication to the SAP Target System
- SAP Administrator Account Permissions

9.1.2.1 Secure Communication to the SAP Target System

Secure communication between Oracle Waveset and the SAP target system is provided by SAP Secure Network Communications (SNC).

For more information about SNC, see the following sections:

- SAP Secure Network Communications (SNC) Parameters
- Configuring Secure Network Communications (SNC) for the SAP User Management Connector

For general information about SNC, see the following article:

`http://help.sap.com/saphelp_nw73/helpdata/en/0a/0a2e0fef6211d3a6510000e835363f/content.htm`

9.1.2.2 SAP Administrator Account Permissions

To manage the user accounts on the SAP target system, the SAP administrator must have account permissions to create and modify user accounts, including read, write, and delete permissions.

For information about configuring the SAP administrator, see SAP Administrator Credentials Parameters.

9.1.3 Certified Components for the SAP User Management Connector

The SAP User Management connector is certified with the components shown in the following table.

Table 9–8 Certified Components for the SAP User Management Connector

Component	Requirement
Oracle Waveset	Oracle Waveset 8.1 Update 1 Bundle Patch 8 or later
SAP target systems	The following SAP target systems are supported: ■ SAP R/3 4.7 SP 45 (running on WAS 6.20) BASIS SP 48 or later ■ mySAP ERP 2004 (ECC 5.0 running on WAS 6.40) BASIS SP 22 or later ■ mySAP ERP 2005 (ECC 6.0 running on WAS 7.00) BASIS SP 13 or later Note: From version 6.40 onward, SAP WAS is also known as SAP NetWeaver. ■ SAP NetWeaver 7.0 with SAP BASIS 7.00 and SAP Business Suite release: BS 2005 with the following constituents: – SAP ERP 6.0 (EHP2 and EHP3) – SAP CRM 5.0, 6.0 – SAP SRM 5.0, 6.0 – SAP SCM 5.0, 5.1 ■ SAP NetWeaver 7.0 EHP2 with SAP BASIS 7.20 and SAP Business Suite release: BS 7i 2010 with the following constituents: – SAP ERP 6.0 EHP 5 (EHP 5) – SAP CRM 7.0 EHP1 – SAP SRM 7.0 EHP1 – SAP SCM 7.0 EHP1
SAP target systems (continued)	■ SAP NetWeaver 7.0 EHP1 with SAP BASIS 7.01 and SAP Business Suite release: BS 2007 with the following constituents: – SAP ERP 6.0 EHP 4 (EHP 4) – SAP CRM 7.0 – SAP SRM 7.0 – SAP SCM 7.0 ■ SAP NetWeaver 7.0 EHP3 with SAP BASIS 7.31 and SAP Business Suite release: BS 7i 2011 with the following constituents: – SAP ERP 6.0 EHP 6 (EHP 6) – SAP CRM 7.0 EHP2 – SAP SRM 7.0 EHP2 – SAP SCM 7.0 EHP2 In general: ■ SAP applications installed on the ABAP stack are supported. ■ Applications installed on the JAVA stack are not supported. ■ Some SAP applications can be installed on the ABAP+JAVA stack. While installing such an application, you specify either ABAP or JAVA as the data source. The connector supports SAP applications that use the ABAP data source. ■ SAP applications and modules that support user management using transaction code SU01 are supported.

Table 9–8 (Cont.) Certified Components for the SAP User Management Connector

Component	Requirement
Identity Connector Framework (ICF)	ICF 1.0 or later
External Code	SAP Java Connector (SAP JCo) 3.0.2 or later, including: ■ For all platforms: sapjco3.jar ■ For Microsoft Windows platforms: sapjco3.dll ■ For UNIX and Linux platforms: libsapjco3.so
JDK	JDK 1.5 or later

9.1.4 Supported Languages for the SAP User Management Connector

The SAP User Management connector is localized in the following languages:

- Arabic
- Chinese (Simplified and Traditional)
- Czech
- Danish
- Dutch
- Finnish
- French
- German
- Greek
- Hebrew
- Hungarian
- Italian
- Japanese
- Korean
- Norwegian
- Polish
- Portuguese (Brazilian)
- Romanian
- Russian
- Slovak
- Spanish
- Swedish
- Thai
- Turkish

9.2 Deploying the SAP User Management Connector

You can deploy the SAP User Management connector either locally in Oracle Waveset or remotely in the Connector Server, as described in the following sections:

- Downloading and Installing the SAP Java Connector (JCo) Files
- Installing the SAP User Management Connector in the Connector Server
- Installing the SAP User Management Connector in Oracle Waveset
- Postinstallation Tasks for the SAP User Management Connector

9.2.1 Downloading and Installing the SAP Java Connector (JCo) Files

The SAP User Management connector requires the following SAP JCo files:

- For all platforms: sapjco3.jar
- For Microsoft Windows platforms: sapjco3.dll
- For UNIX and Linux platforms: libsapjco3.so

These files are available in the SAP JCo.zip file. To download and install these files, perform the following steps on the Oracle Waveset host computer:

1. Download the JCo.zip file from the SAP site as follows:
   a. Select **Application Platform, Connectivity, Connectors, SAP Java Connector,** and **Tools & Services** to open the SAP JAVA Connector page.
   b. On the SAP JAVA Connector page, in the right pane, click the link for the SAP JCo release that you want to download.
   c. In the dialog box that is displayed, specify the path to the directory in which you want to save the file and click **Save.**

   For the versions of the SAP JCo that are supported, see Certified Components for the SAP User Management Connector.
2. Create a new directory and then extract the JCo.zip file in this new directory.
3. Depending on your platform, copy the SAP library files to the following directories:
   - For Windows platforms:
     a. Copy sapjco3.dll to the winnt\system32 directory. Or, copy this file into any directory and then add the path to the directory to the PATH environment variable.
     b. Ensure that the msvcr80.dll and msvcp80.dll files are in the C:\WINDOWS\system32 directory. If necessary, first download these files from various sources on the Internet.
   - For UNIX and Linux platforms:
     a. Copy libsapjco3.so to the /usr/local/jco directory.
     b. Add the path to the directory specified in the previous step to the LD_LIBRARY_PATH environment variable.
4. Stop the Oracle Waveset web application.
5. Copy the SAP User Management connector bundle JAR file (org.identityconnectors.sap-2.0.0.jar) to the

WavesetInstallDirectory/WEB-INF/bundles directory of the Oracle Waveset web application.

In the JAR file name, 2.0.0 represents the connector bundle version.

6. Copy the sapjco3.jar file to the *WavesetInstallDirectory*/WEB-INF/lib directory.
7. Copy the sap-idmglue.jar file from the sap-idmglue-2.0.0/WEB-INF/lib directory to the *WavesetInstallDirectory*/WEB-INF/lib directory.
8. Start the Oracle Waveset web application.
9. Import the following XML files from the sap-idmglue-2.0.0/sample/connectors/sap-idmglue directory:
   - postProcess.xml
   - Resource Wizard SAPConnector Connector.xml
   - SAP Connector CUA User Form.xml
   - SAP Connector User Form.xml

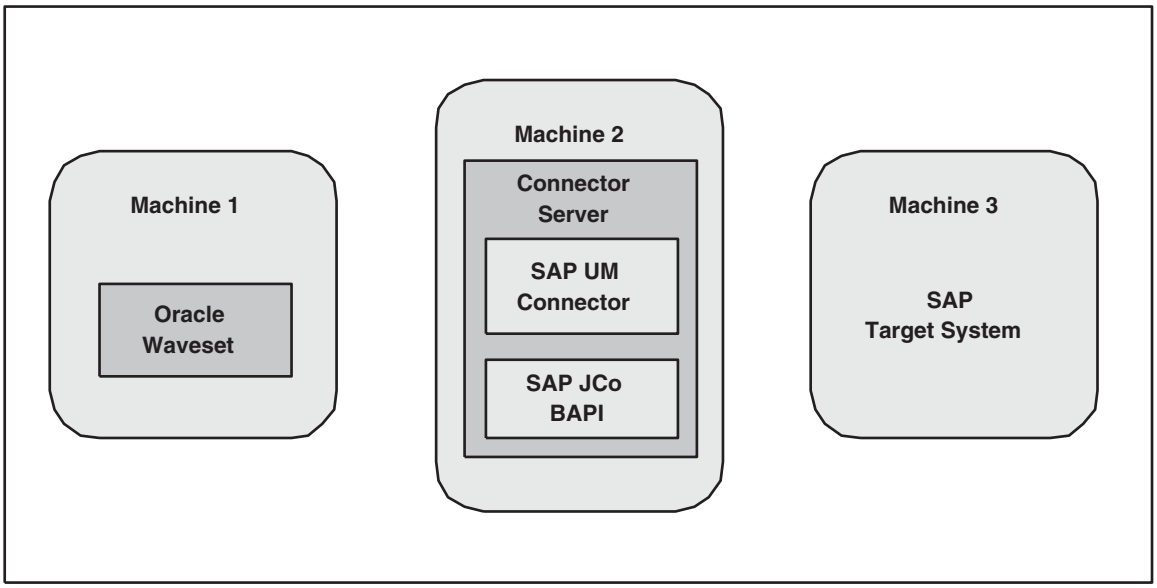
9.2.2 Installing the SAP User Management Connector in the Connector Server

This section describes the following subsections:

9.2.2.1 SAP User Management Connector Deployment Architecture With the Connector Server

The following figure shows a distributed deployment architecture with the SAP User Management connector deployed in the Connector Server.

Figure 9–2 SAP User Management Connector Deployment Architecture With the Connector Server



A SAP User Management connector deployment with the Connector Server includes these components:

- **Machine 1** has Oracle Waveset deployed.
- **Machine 2** has the SAP User Management connector installed in the Connector Server. The Connector Server is part of the Identity Connector Framework (ICF).

  The SAP User Management connector uses the Business Application Programming Interface (BAPI) to send requests to the SAP target system. The BAPI is available in the SAP Java Connector (JCo) package.
- **Machine 3** has the SAP target system deployed.

9.2.2.2 Installing and Configuring the Connector Server

Note: The Connector Server requires a JDK to run. For the requirements, see Certified Components for the SAP User Management Connector. If necessary, set your JAVA_HOME environment variable to point to your specific installation.

To install and configure the Connector Server:

1. Create a new directory on the machine where you want to install the Connector Server. In this section, `CONNECTOR_SERVER_HOME` represents this directory.
2. Unzip the Connector Server package in your new directory from Step 1. The Connector Server package is available with the Identity Connector Framework (ICF).
3. In the `ConnectorServer.properties` file, set the following properties, as required by your deployment. The `ConnectorServer.properties` file is located in the `conf` directory.

Property	Description
connectorserver.port	Port on which the Connector Server listens for requests. The default is 8759.
connectorserver.bundleDir	Directory where the connector bundles are deployed. The default is bundles.
connectorserver.libDir	Directory in which to place dependent libraries. The default is lib.
connectorserver.usessl	If set to true, the Connector Server uses SSL for secure communication. The default is false. If you specify true, use the following options on the command line when you start the Connector Server: ■ -Djavax.net.ssl.keyStore ■ -Djavax.net.ssl.keyStoreType (optional) ■ -Djavax.net.ssl.keyStorePassword
connectorserver.ifaddress	Bind address. To set this property, uncomment it in the file (if necessary). The bind address can be useful if there are more NICs installed on the machine.
connectorserver.key	Connector Server key.

4. Set the properties in the ConnectorServer.properties file, as follows:
   - To set connectorserver.key, run the Connector Server with the setKey option.
   - For all other properties, edit the ConnectorServer.properties file manually.
5. The conf directory also contains the logging.properties file, which you can edit if required by your deployment.

9.2.2.3 Running the Connector Server on Windows Systems

To run the Connector Server on Windows systems, use the ConnectorServer.bat script as follows:

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

   The ConnectorServer.bat script supports the following options:

Option	Description
/install [*serviceName*] ["-J *java option*"]	Installs the Connector Server as a Windows service. Optionally, you can specify a service name and Java options. If you do not specify a service name, the default name is ConnectorServerJava.

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

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

9.2.2.4 Running the Connector Server on UNIX and Linux Systems

To run the Connector Server on UNIX and Linux systems, use the `connectorserver.sh` script, as follows:

1. Make sure that you have set the properties required by your deployment in the `ConnectorServer.properties` file, as described in Installing and Configuring the Connector Server.
2. Change to the `CONNECTOR_SERVER_HOME/bin` directory.
3. Use the `chmod` command to set the permissions to make the `connectorserver.sh` script executable.
4. Run the `connectorserver.sh` script. The script supports the following options:

Option	Description
`/run [ -Jjava-option ]`	Runs the Connector Server in the console. Optionally, you can specify one or more Java options. For example, to run the Connector Server with SSL: `./connectorserver.sh /run -J-Djavax.net.ssl.keyStore=mykeystore.jks -J-Djavax.net.ssl.keyStorePassword=password`
`/start [ -Jjava-option ]`	Runs the Connector Server in the background. Optionally, you can specify one or more Java options.
`/stop`	Stops the Connector Server, waiting up to 5 seconds for the process to end.
`/stop n`	Stops the Connector Server, waiting up to *n* seconds for the process to end.
`/stop -force`	Stops the Connector Server. Waits up to 5 seconds and then uses the `kill -KILL` command, if the process is still running.
`/stop n -force`	Stops the Connector Server. Waits up to *n* seconds and then uses the `kill -KILL` command, if the process is still running.

Option	Description
`/setKey key`	Sets the Connector Server key. The `connectorserver.sh` script stores the hashed value of *key* in the `connectorserver.key` property in the `ConnectorServer.properties` file.

9.2.2.5 Installing the SAP User Management Connector

To install the SAP User Management connector in the Connector Server:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the SAP User Management Connector.
2. Make sure you have performed the Downloading and Installing the SAP Java Connector (JCo) Files.
3. Stop the Connector Server.
4. Copy the SAP User Management connector bundle to the `CONNECTOR_SERVER_HOME/bundles` directory.
5. Copy the sapjco3.jar file to the CONNECTOR_SERVER_HOME/lib directory.
6. Start the Connector Server.

For information about starting and stopping the Connector Server, see Running the Connector Server on Windows Systems or Running the Connector Server on UNIX and Linux Systems.

Continue with Postinstallation Tasks for the SAP User Management Connector.

9.2.3 Installing the SAP User Management Connector in Oracle Waveset

To install the SAP User Management connector in Oracle Waveset:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the SAP User Management Connector.
2. Make sure you have performed the Downloading and Installing the SAP Java Connector (JCo) Files.
3. Stop the Oracle Waveset web application.
4. Copy the SAP User Management bundle JAR file to the *WavesetInstallDirectory*/WEB-INF/bundles directory.
5. Copy the sapjco3.jar file to the *WavesetInstallDirectory*/WEB-INF/lib directory.
6. Start the Oracle Waveset web application.

Continue with Postinstallation Tasks for the SAP User Management Connector.

9.2.4 Postinstallation Tasks for the SAP User Management Connector

After you install the SAP User Management connector, perform the following tasks:

- Creating a SAP User Management Connector Resource
- Configuring Secure Network Communications (SNC) for the SAP User Management Connector
- Enabling the Use of a Logon Group for the SAP User Management Connector
- Enabling SAP JCo Connectivity for the SAP User Management Connector

9.2.4.1 Creating a SAP User Management Connector Resource

To create an SAP User Management connector resource:

1. Log in to the Oracle Waveset Administrator interface.
2. Create the SAP User Management connector resource by following the Create SAP User Management Connector Resource wizard.
3. Select the SAP User Management Connector Version as "2.0.0".
4. If the SAP User Management connector is deployed in the Connector Server, select the Connector Server on which the connector bundle is deployed.

   Or, if the SAP User Management connector is deployed in Oracle Waveset, specify the value for the Java Connector Server as Local.
5. Specify values for the SAP User Management connector, depending on your deployment. For more information, see:
   - SAP User Management Connector Resource Configuration Parameters
   - SAP User Management Connector Account Attributes

9.2.4.2 Configuring Secure Network Communications (SNC) for the SAP User Management Connector

Oracle Waveset uses a Java application server. To connect to the SAP system application server, this Java application server uses the SAP Java connector (JCo). If required, you can use Secure Network Communication (SNC) to secure communication between Oracle Waveset and the SAP target system.

This section describes the following topics:

- Prerequisites for Configuring the SAP User Management Connector to Use SNC
- Installing the Security Package
- Configuring SNC

9.2.4.2.1 Prerequisites for Configuring the SAP User Management Connector to Use SNC The following are prerequisites for configuring the SAP User Management connector to use SNC:

- SNC must be activated on the SAP application server.
- You must be familiar with the SNC infrastructure. You must know which Personal Security Environment (PSE) the application server uses for SNC.

9.2.4.2.2 Installing the Security Package To install the security package on the Java application server used by Oracle Waveset:

1. Extract the contents of the SAP Cryptographic Library installation package.

   The SAP Cryptographic Library installation package can be ordered from SAP official software partners listed on the SAP site.

   The security package contains the following files:
   - SAP Cryptographic Library:
     - Microsoft Windows platforms: sapcrypto.dll
     - UNIX and Linux platforms: libsapcrypto.so
   - A corresponding license ticket (ticket)

- The configuration tool:
  - Microsoft Windows platforms: sapgenpse.exe
  - UNIX and Linux platforms: sapgenpse

2. Copy the library and the sapgenpse.exe or sapgenpse file to a local directory. For example, on Windows:

   ```
   C:/usr/sap
   ```

3. Check the file permissions. Ensure that the user under which the Java application server runs is able to run the library functions in the directory into which you copied the library and the sapgenpse.exe file.
4. Create the sec directory inside the directory into which you copied the library and the sapgenpse.exe file.

   You can use any names for the directories that you create. However, creating the `C:\usr\sap\sec` or `/usr/sap/sec` directory is the SAP recommendation.
5. Copy the ticket file into the sec directory. This is also the directory in which the Personal Security Environment (PSE) and credentials of the Java application server are generated.
6. Set the SECUDIR environment variable for the Java application server user to the sec directory.

   From this point onward, the term SECUDIR directory is used to refer to the directory whose path is defined in SECUDIR environment variable.
7. Set the SNC_LIB and PATH environment variables for the user of the Java application server to the cryptographic library directory, which is the parent directory of the sec directory.

9.2.4.2.3 Configuring SNC To configure SNC for the SAP User Management connector:

1. Either create a PSE or copy the SNC PSE of the SAP application server to the SECUDIR directory. To create the SNC PSE for the Java application server, use the sapgenpse.exe command-line tool as follows:
   a. To determine the location of the SECUDIR directory, run the sapgenpse command without specifying any command options. The program displays information such as the library version and the location of the SECUDIR directory.
   b. Enter a command similar to the following to create the PSE:

      ```
      sapgenpse get_pse -p PSE_Name -x PIN Distinguished_Name
      ```

      The following is a sample distinguished name:

      ```
      CN=SAPJ2EE, O=MyCompany, C=US
      ```

      The sapgenpse command creates a PSE in the SECUDIR directory.
2. Create credentials for the Java application server.

   The Java application server must have active credentials at run time to be able to access its PSE. To check whether or not this condition is met, enter the following command in the parent directory of the SECUDIR directory:

   ```
   sapgenpse seclogin
   ```

Then, enter the following command to open the PSE of the server and create the credentials.sapgenpse file:

```
seclogin -p PSE_Name -x PIN -O [NT_Domain\]user_ID
```

The user_ID that you specify must have administrator rights. PSE_NAME is the name of the PSE file.

The credentials file, cred_v2, for the user specified with the -O option is created in the SECUDIR directory.

3. Exchange the public key certificates of the two servers as follows:

   a. Export the Oracle Waveset certificate by entering the following command:

   ```
   sapgenpse export_own_cert -o filename.crt -p PSE_Name -x PIN
   ```

   b. Import the Oracle Waveset certificate into the SAP application server. You might require the SAP administrator's assistance to perform this step.

   c. Export the certificate of the SAP application server. You may require the SAP administrator's assistance to perform this step.

   d. Import the SAP application server certificate into Oracle Waveset by entering the following command:

   ```
   sapgenpse maintain_pk -a serverCertificatefile.crt -p PSE_Name -x PIN
   ```

4. Configure the following parameters:

   - Enable SAP SNC
   - SNC Protection Level
   - SNC Name
   - SNC Partner Name
   - SNC X509 Certificate
   - SNC Library Path

   For a description of these parameters, see SAP Secure Network Communications (SNC) Parameters.

9.2.4.3 Enabling the Use of a Logon Group for the SAP User Management Connector

In SAP, a logon group is used for failover and as a load-sharing mechanism. When a user logs in to a logon group, the system internally routes the connection request to the logon group member with the least load.

To enable the use of a logon group, set the following SAP User Management connector resource configuration parameters:

- App server host
- Logon group name
- Message server
- R3 name

9.2.4.4 Enabling SAP JCo Connectivity for the SAP User Management Connector

Perform the following steps either on the Oracle Waveset host computer or the Connector Server, depending on where you deployed the SAP User Management connector.

To enable SAP JCo connectivity:

1. Open the following file in a text editor:
   - For Microsoft Windows platforms:

     ```
     C:\WINDOWS\system32\drivers\etc\services
     ```

   - For UNIX or Linux platforms:

     ```
     /etc/services
     ```

2. Add an entry to the file from the previous step in the following format:

   sapms*SYSTEM_ID* 36*SYSTEM_NUMBER*/tcp

   For example, the new entry is shown in bold text:

   ```
   ...
   ipx 213/udp #IPX over IP
   ldap 389/tcp #Lightweight Directory Access Protocol
   sapmsE60 3600/tcp
   ```

3. Save and close the file.
4. Create the sapmsg.ini file and add the following lines in the file:

   ```
   [Message Server]
   o01=oss001.wdf.sap-ag.de
   SYSTEM_ID=HOST_NAME
   ```

   For example:

   ```
   [Message Server]
   o01=oss001.wdf.sap-ag.de
   E60=mysap08.corp.example.com
   ```

5. Save and close the file.
6. Copy the sapmsg.ini file to the C:\ directory for Windows systems or the root directory for UNIX and Linux systems.

9.3 Using the SAP User Management Connector

This section provides the following information:

- SAP User Management Connector Account Attributes
- Sample Forms for the SAP User Management Connector

9.3.1 SAP User Management Connector Account Attributes

The following table lists the SAP User Management connector account attributes. These attributes are in the User object class. The only required attributes are accoundId and lastname.

Table 9–9 SAP User Management Connector Account Attributes

Account Attribute	Description
accountId	User's account ID. Required.
firstname	User's first name.
fullname	User's full name.
email	User's email address.
lastname	User's last name. Required.
groups	Provisions to the SAP GROUPS table.
accountLockedNoPwd	Boolean attribute that indicates whether the account is locked because the user has no password.
accountLockedWrngPwd	Boolean attribute that indicates whether the account is locked because of failed login attempts.
personNumber	Internal key for identifying a person.
addressNumber	Internal key for identifying an address for central address management.
birthName	Maiden name or name given at birth.
middleName	User's middle name.
secondLastName	User's second last name.
academicTitle	User's academic title, such as Dr. or Prof.
academicTitle2	Second academic title for the user.
namePrefix	User's prefix to a last name, such as von, van der, or de la.
namePrefix2	Second prefix to the user's last name.
titleSupplement	Name supplement, for the user. For example, a noble title, such as Lord or Lady.
nickname	User's nickname.
initials	User's middle initial or initials.
nameFormat	Sequence in which name components are assembled to present the name of a person in a complete form. The sequence can vary for each country.
nameFormatCountry	Country used to determine the name format.
languageKey	Language used to enter and display text.
iso639Language	ISO 639 language code.
sortKey1	Search term.
sortKey2	Secondary search term.
department	Department in a company as part of the company address
function	User's job functionality.
buildingNumber	Building number where the user's office is located.
buildingFloor	Floor where the user's office is located.
correspondenceCode	Correspondence code.
inhouseMailCode	Internal mail code.
communicationType	States how the user wants to exchange documents and messages with a business partner.
title	Title such as Mr. or Mrs.

Table 9–9 (Cont.) SAP User Management Connector Account Attributes

Account Attribute	Description
titleP	Title such as Mr. or Mrs.
addressName	Name of an address.
addressName2	Second line in the name of an address.
addressName3	Third line in the name of an address.
addressName4	Fourth line in the name of an address.
careOfName	Part of the address if the recipient is different from the occupant (c/o = care of) .
city	User's city
district	City or district supplement.
cityNumber	City code.
districtNumber	District code.
cityPostalCode	User's postal code.
poBoxPostalCode	Postal code required for unique assignment of the Post Office box.
companyPostalCode	Postal code that is assigned directly to a company.
poBox	User's post office box.
poBoxCity	Post office box city.
poBoxCityCode	Post Office box city, if it is different from the address city.
postalDeliveryDistricy	Postal delivery district.
transportZone	Regional zone of a goods recipient or supplier.
street	User's street.
streetNumber	User's street code.
streetAbbreviation	User's street abbreviation.
houseNumber	Number portion of a street address.
houseNumber2	Secondary addess number
street2	Additional address field printed above the street line.
street3	Additional address field printed above the street line.
street4	Additional address field printed above the street line.
street5	Additional address field printed above the street line.
oldBuilding	Number or ID for the building in a contact person address.
floor	Floor number of an address.
roomNumber	Room number in an address.
countryCode	Country in an address.
countryCodeISO	Two-letter ISO code for the country in an address.
languageKey	Language used to enter and display text.
languageKeyISO	ISO 639 language code.
region	State or province.
sort2	Secondary search term.
timeZone	Time difference of the time zone in hours/minutes relative to the UTC.

Table 9–9 (Cont.) SAP User Management Connector Account Attributes

Account Attribute	Description
taxJurisdictionCode	Tax authority to which taxes must be paid. It is always the city to which the goods were delivered.
telephoneNumber	Telephone number, including the area code, but no country code.
telephoneExtension	Telephone number extension.
faxNumber	Fax number, including the area code, but no country code.
faxExtension	Fax number extension.
cuaSystems	Central User Administration system names.
profiles	Profiles assigned to the user.
activityGroups	Roles assigned to the user.
lastLoginTime	Read-only attribute that lists the most recent login time.

9.3.2 Sample Forms for the SAP User Management Connector

The SAP User Management connector includes the following forms:

- SAP Connector User Form
- SAP Connector CUA User Form

The connector also includes the Resource Wizard SAP Connector.xml and postProcess.xml files.

9.4 Troubleshooting the SAP User Management Connector

Use the Oracle Waveset debug pages to set trace options on the following class:

```
org.identityconnectors.sap
```

This class returns the available error messages from the SAP target resource.

9.5 Known Issues for the SAP User Management Connector

The SAP User Management connector has the following known issue.

- Multi-valued Attributes Prefixed with Underscore
- Class Loader Issue with the SAP User Management Connector

9.5.1 Multi-valued Attributes Prefixed with Underscore

SAP Resource Adapter supports Role, Profile, Group, and Parameter multi-valued attributes. However, it does not support multi-valued attributes prefixed with underscore ("_").

In Oracle Waveset 8.1.1.7, some functions and constants have been deprecated in the SAPResourceAdapter class. The PARAMETER->PARAMOBJ multivalued attribute is of the type complex and is not a simple string. Therefore, multi-valued attributes prefixed with underscore ("_") are not supported in SAP Resource Adapters.

9.5.2 Class Loader Issue with the SAP User Management Connector

The SAP JCo must register the data provider with the JCo Environment class with a destination name. Any number of destination names can be added to the provider.

The Identity Connector Framework (ICF) uses a different class loader for each bundle, so if two SAP connector bundles (such as SAP User Management and SAP HR) are installed, then the connector bundle that creates a connection first will work.

However, if a second SAP connector bundle tries to create a connection, it tries to register the data provider, which is already registered by the first SAP connector bundle. It then throws the "DestinationDataProvider already registered" error.

Consider the following scenarios:

- **Scenario 1** - Two connectors: SAP User Management connector and SAP HR connector

  A different class loader is used for the SAP User Management connector and the SAP HR connector. For example:

  1. Create Resource 1 for the SAP User Management connector. The resource will create successfully.
  2. Create Resource 2 for the SAP User Management connector. Because the provider is static, it is already created for Resource 1. Therefore, this step will not register the provider again. It will add the destination to the existing provider, and the resource creation will be successful.
  3. Create Resource 1 for the SAP HR connector. Because this connector uses a different class loader, this step will try to register a new provider because the provider instance will not be available in the new class loader. It will then throw the exception "Provider is already registered" because only one provider can be registered in the JCo Environment.

     The JCODestinationManager will try to get the destination of the SAP HR connector from the Environment. In the Environment, destinations of the SAP User Management connector will be available. Therefore, it will throw the exception "Destination does not exist", and the Resource creation will fail.

  In this scenario, you can create any number of resources for the SAP User Management connector, but any attempts at resource creation for the SAP HR connector will fail.

  However, if you create a resource for the SAP HR connector first after starting the application server, then any number of resources for the SAP HR connector will work. But resource creation for the SAP User Management connector will fail.

- **Scenario 2** - One connector and one Resource Adapter: SAP connector and SAPJco3HRActiveSyncAdapter

  When the SAP connector is used, it will have a different class loader. Therefore, the behavior will be similar to Scenario 1.

- **Scenario 3** - Two Resource Adapters: SAPBasisResourceAdapter and SAPJco3HRActiveSyncAdapter

  The same class loader is used for both Resource Adapters. It will create a new DestinationDataProvider when the first resource is created (either SAP User Management or SAP HR). It will use the same provider for all subsequent resource creations whether it is the SAP User Management Resource Adapter or SAP HR Resource Adapter. Therefore, all resource creations will be successful.

  The issue occurs only if a connector is used because each connector uses a different class loader. Resource Adapters do not have this issue because they use the same class loader.

To use the SAP User Management connector and a Resource Adapter, deploy the SAP User Management connector in the Connector Server and the Resource Adapter in Oracle Waveset.

10

Oracle Waveset Connector for Siebel User Management

This chapter includes the following information about the Siebel connector for Oracle Waveset:

10.1 About the Siebel Connector

10.1.1 Overview of the Siebel Connector

Oracle Waveset communicates with the Siebel target system through the Siebel connector using the Siebel target APIs. The Siebel Business Applications environment consists of entities such as Siebel Enterprise Server, Siebel Gateway Name Server, Siebel Database Server, and Siebel File System. For more information, see the Siebel Applications documentation:

`http://www.oracle.com/technetwork/indexes/documentation/index.html`

The Siebel connector is implemented using the Identity Connector Framework (ICF). The ICF provides a container that separates the connector bundle from the application. The ICF also provides common features that developers would otherwise need to

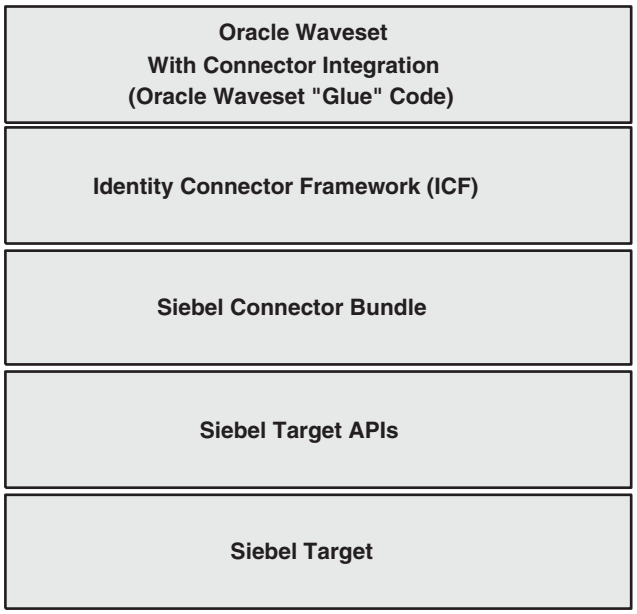
implement on their own, such as connection pooling, buffering, time outs, and filtering. For more information about the ICF, see Chapter 1, "Identity Connectors Overview".

The Siebel connector supersedes the Siebel resource adapter. To migrate a Siebel resource adapter deployment, see Migrating a Siebel Resource Adapter.

10.1.1.1 Siebel Connector Architecture

The following figure shows the Siebel connector architecture.

Figure 10–1 Siebel Connector Architecture



The Siebel connector architecture includes these components:

- Oracle Waveset includes the connector integration files. These files are XML files that provide the configuration information necessary to transform data from a resource to Oracle Waveset. Integration files are sometimes called the connector "glue" code.
- The Identity Connector Framework (ICF) provides a container that separates the connector bundle from the application. The ICF also provides common features that developers would otherwise need to implement on their own, such as connection pooling, buffering, time outs, and filtering.
- The Siebel connector bundle uses the Siebel target APIs to access the Siebel target system.

10.1.1.2 Siebel Connector Features

The Siebel connector supports the following features:

- Create account
- Update account
- Delete account
- Full reconciliation

10.1.1.3 Siebel Connector Resource Object Management

By default, the Siebel connector supports the following Siebel object:

Resource Object	Features Supported	Attributes Managed
Account (__ACCOUNT__ object class)	■ Create ■ Update ■ Delete ■ Full reconciliation	Login Name, First Name, Last Name, Middle Name, Work Phone Extension, EMail Addr, Alias, Employee Type Code, Time Zone, Preferred Communications, Job Title, Home Phone #, Personal Title, Phone #, Fax #, Responsibility, Position

10.1.1.4 Siebel Connector Configuration Parameters

The Siebel connector supports the following configuration parameters.

Table 10–1 Siebel Connector Configuration Parameters

Parameter	Description
Enterprise Server	Name of the Enterprise server. An Enterprise is a logical collection of Siebel servers that access a single database server and file system. Sample value: siebel
Gateway Server	Name of the Gateway server. A Gateway server is a Windows service or UNIX daemon process that stores component definitions and assignments, operational parameters, and connectivity information. Sample value: SBA_SIEBEL
Gateway Server Port	Listening port number for the Siebel Connection Broker (alias SCBroker). Sample value : 2321
Language	Language in which the text on the UI is displayed. Specify any one of the following values: ■ For English: ENU ■ For Brazilian Portuguese: PTB ■ For French: FRA ■ For German: DEU ■ For Italian: ITA ■ For Japanese: JPN ■ For Korean: KOR ■ For Simplified Chinese: CHS ■ For Spanish: ESP ■ For Traditional Chinese: CHT

Table 10–1 (Cont.) Siebel Connector Configuration Parameters

Parameter	Description
Object Manager	Name of the object manager. Specify any one of the following values: ■ For English: SCCObjMgr_enu ■ For Brazilian Portuguese: SCCObjMgr_ptb ■ For French: SCCObjMgr_fra ■ For German: SCCObjMgr_deu ■ For Italian: SCCObjMgr_ita ■ For Japanese: SCCObjMgr_jpn ■ For Korean: SCCObjMgr_kor ■ For Simplified Chinese: SCCObjMgr_chs ■ For Spanish: SCCObjMgr_esp ■ For Traditional Chinese: SCCObjMgr_cht
Password	Password of the target system user account that you want to use for connector operations. Sample value: sadmin
Siebel Server	Name of the target system server. Sample value: SBA_SIEBEL
User Name	User ID of the target system user account that you want to use for connector operations. Sample value: SADMIN
Encryption	Type of encryption for secure communication. If encryption is required, then specify RSA. Otherwise, specify None. Note: The value of this parameter is case-sensitive. Default value: None
Siebel Version	Version of the target system supported by this connector. Sample value: 8.1.1
SSO Flag	Enter Yes to specify that the target system is configured to use an SSO solution for authentication. Otherwise, enter No. Default value: No
Employee Business Object	Business Object of Employee userType. Default value: Employee
Employee Business Component	Business Component of Employee `userType`. Default value: Employee
User Business Object	Business Object of the User `userType`. Default value: Users
User Business Component	Business Component of the User `userType`. Default value: User
Trusted Token	Enter the trusted token value that you specify while configuring the target system to communicate with the SSO system. If you have not configured SSO authentication, then enter No.
Key Field Name	Enter the search attribute in the Siebel Business Component that must be treated as the unique identifier for an account. The format of this parameter is as follows: `ATTRIBUTE_TYPE;ATTRIBUTE_NAME` Default value: `common;Login Name`

10.1.2 Requirements for the Siebel Connector

The Siebel connector for Oracle Waveset has the following requirements:

- Oracle Waveset with the patch shown in Certified Components for the Siebel Connector must be installed.

- If you are installing the Siebel connector in Oracle Waveset, the following Siebel JAR files are required in the `$WSHOME/WEB-INF/lib` directory of the Oracle Waveset web application:
  - Siebel 7.8 through 8.1.1: `Siebel.jar` and `SiebelJI_enu.jar`
  - Siebel 7.5 through 7.7: `SiebelJI_Common.jar`, `SiebelJI_enu.jar`, and `SiebelJI.jar`

  The Siebel JAR files are available in the `SIEBEL_INSTALLATION_DIRECTORY/siebsrvr/CLASSES` directory.

  Note. To prevent potential conflicts, do not copy JAR files for multiple versions of Siebel into the `$WSHOME/WEB-INF/lib` directory.
- If you are installing the Siebel connector in the Java Connector Server, the Siebel JAR files are required in the `CONNECTOR_SERVER_HOME/lib` directory.

10.1.3 Security Considerations for the Siebel Connector

This section provides the following security information for the Siebel connector:

- **Supported Connections**

  Oracle Waveset can use HTTP or RSA encryption to communicate with the Siebel connector. For more information, see the Siebel Applications documentation:

  `http://www.oracle.com/technetwork/indexes/documentation/index.html`

- **Required Administrative Privileges**

  The administrator user name and password configured for the Siebel connector must be assigned sufficient privileges within Siebel to create new records and to update existing records for the specified business component.

10.1.4 Certified Components for the Siebel Connector

The Siebel connector is certified with the following components:

Table 10–2 Certified Components for the Siebel Connector

Component	Requirement
Oracle Waveset	Oracle Waveset 8.1.1 Patch 5 or later
Identity Connector Framework (ICF)	ICF 1.1 or later
Target Systems	▪ Oracle Siebel 7.8 through 8.1.1 ▪ Oracle Siebel 7.5 through 7.7
JDK	JDK 1.5 or later

10.1.5 Supported Languages for the Siebel Connector

The Siebel connector is localized in the following languages:

- Arabic
- Chinese (Simplified)
- Chinese (Traditional(
- Danish

- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazilian)
- Spanish

10.2 Migrating a Siebel Resource Adapter

If you currently have the Siebel resource adapter installed, this section describes how to migrate to the Siebel connector.

To migrate a Siebel resource adapter, follow these steps:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the Siebel Connector.
2. Log in to the Oracle Waveset Administrator interface.
3. Select the Resources tab and then the Migrate Adapters tab.
4. Follow the Migration Wizard and select the Siebel Resource adapter and corresponding Siebel connector.

10.3 Deploying the Siebel Connector

You can deploy the Siebel connector either locally in Oracle Waveset or remotely in the Java Connector Server, as described in the following sections:

- Installing the Siebel Connector in Oracle Waveset
- Deploying the Siebel Connector in the Java Connector Server
- Creating a Siebel Connector Resource

10.3.1 Installing the Siebel Connector in Oracle Waveset

To install the Siebel connector in Oracle Waveset, follow these steps:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the Siebel Connector.
2. Stop the Oracle Waveset web application.
3. Copy the following Siebel connector JAR files into the `$WSHOME/WEB-INF/lib` directory of the Oracle Waveset web application:
   - Siebel 7.8 through 8.1.1: `Siebel.jar` and `SiebelJI_enu.jar`
   - Siebel 7.5 through 7.7: `SiebelJI_Common.jar`, `SiebelJI_enu.jar`, and `SiebelJI.jar`

   The Siebel JAR files are available in the `SIEBEL_INSTALLATION_DIRECTORY/siebsrvr/CLASSES` directory.

   Note. To prevent potential conflicts, do not copy JAR files for multiple versions of Siebel into the `$WSHOME/WEB-INF/lib` directory.
4. Start the Oracle Waveset web application.

5. Log in as an Oracle Waveset Administrator.
6. Select the appropriate version of the Siebel connector and create the Siebel connector resource by following the Create Siebel Connector Resource wizard.

   Because you are installing the Siebel connector in Oracle Waveset, specify the value for the Java Connector Server as Local.
7. For the values for the Siebel connector attributes, see Siebel Connector Account Attributes.

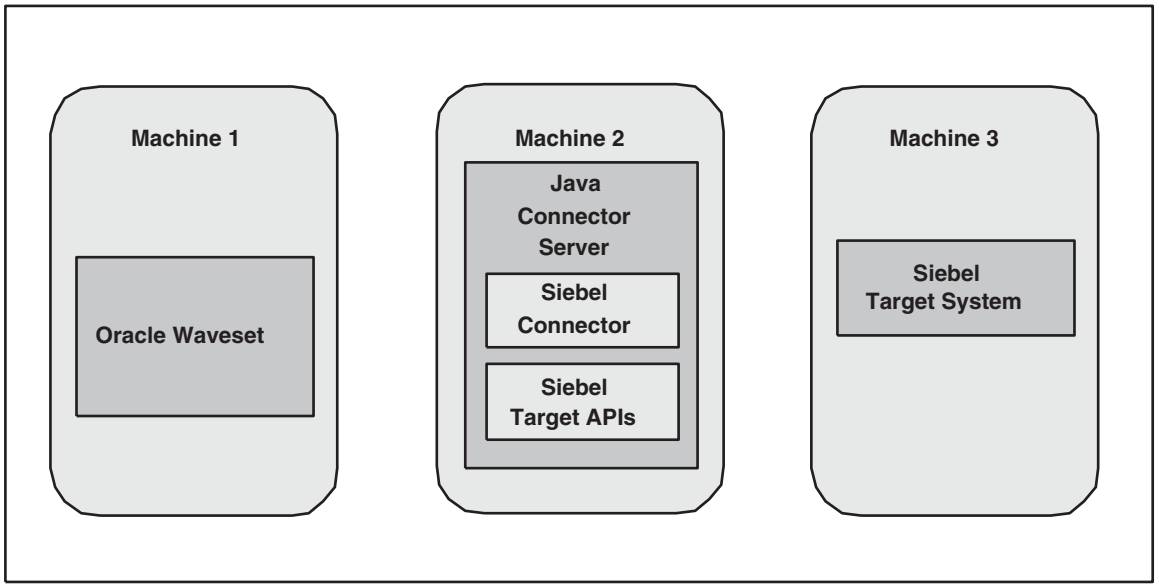
10.3.2 Deploying the Siebel Connector in the Java Connector Server

- Siebel Connector Deployment Architecture With the Java Connector Server
- Installing and Configuring the Java Connector Server
- Running the Java Connector Server on Windows Systems
- Running the Java Connector Server on UNIX and Linux Systems
- Installing the Siebel Connector in the Java Connector Server

10.3.2.1 Siebel Connector Deployment Architecture With the Java Connector Server

If you install the Siebel connector in the Java Connector Server, the following figure shows the distributed deployment architecture.

Figure 10–2 Siebel Connector Deployment Architecture With the Java Connector Server



- **Machine 1** has Oracle Waveset deployed.
- **Machine 2** has the Siebel connector and the Siebel Target APIs (JAR files) installed in the Java Connector Server. The Java Connector Server is part of the Identity Connector Framework (ICF).
- **Machine 3** has the Siebel target system deployed.

10.3.2.2 Installing and Configuring the Java Connector Server

To install and configure the Java Connector Server, follow these steps:

1. Create a new directory on the machine where you want to install the Java Connector Server. In this section, `CONNECTOR_SERVER_HOME` represents this directory.
2. Unzip the Java Connector Server package in your new directory from Step 1. The Java Connector Server package is available with the Identity Connector Framework (ICF).
3. In the `ConnectorServer.properties` file, set the following properties, as required by your deployment. The `ConnectorServer.properties` file is located in the `conf` directory.

Property	Description
`connectorserver.port`	Port on which the Java Connector Server listens for requests. The default is 8759.
`connectorserver.bundleDir`	Directory where the connector bundles are deployed. The default is `bundles`.
`connectorserver.libDir`	Directory in which to place dependent libraries. The default is `lib`.
`connectorserver.usessl`	If set to `true`, the Java Connector Server uses SSL for secure communication. The default is `false`. If you specify `true`, use the following options on the command line when you start the Java Connector Server: ■ `-Djavax.net.ssl.keyStore` ■ `-Djavax.net.ssl.keyStoreType` (optional) ■ `-Djavax.net.ssl.keyStorePassword`
`connectorserver.ifaddress`	Bind address. To set this property, uncomment it in the file (if necessary). The bind address can be useful if there are more NICs installed on the machine.
`connectorserver.key`	Java Connector Server key.

4. Set the properties in the `ConnectorServer.properties` file, as follows:
   - To set `connectorserver.key`, run the Java Connector Server with the option.
   - For all other properties, edit the `ConnectorServer.properties` file manually.
5. The `conf` directory also contains the `logging.properties` file, which you can edit if required by your deployment.

10.3.2.3 Running the Java Connector Server on Windows Systems

To run the Java Connector Server on Windows systems, follow these steps:

1. Make sure that you have set the properties required by your deployment in the `ConnectorServer.properties` file.
2. Change to the `CONNECTOR_SERVER_HOME\bin` directory and find the `ConnectorServer.bat` script.

   The `ConnectorServer.bat` script supports the following options:

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

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

10.3.2.4 Running the Java Connector Server on UNIX and Linux Systems

To run the Java Connector Server on UNIX and Linux systems, use the connectorserver.sh script, as follows:

1. Make sure that you have set the properties required by your deployment in the ConnectorServer.properties file.
2. Change to the CONNECTOR_SERVER_HOME/bin directory.
3. Use the chmod command to set the permissions to make the connectorserver.sh script executable.
4. Run the connectorserver.sh script. The script supports the following options:

Option	Description
/run [-J*java-option*]	Runs the Java Connector Server in the console. Optionally, you can specify one or more Java options. For example, to run the Java Connector Server with SSL: ./connectorserver.sh /run -J-Djavax.net.ssl.keyStore=mykeystore.jks -J-Djavax.net.ssl.keyStorePassword=*password*
/start [-J*java-option*]	Runs the Java Connector Server in the background. Optionally, you can specify one or more Java options.
/stop	Stops the Java Connector Server, waiting up to 5 seconds for the process to end.
/stop *n*	Stops the Java Connector Server, waiting up to *n* seconds for the process to end.

Option	Description
`/stop -force`	Stops the Java Connector Server. Waits up to 5 seconds and then uses the `kill -KILL` command, if the process is still running.
`/stop n -force`	Stops the Java Connector Server. Waits up to *n* seconds and then uses the `kill -KILL` command, if the process is still running.
`/setKey key`	Sets the Java Connector Server key. The `connectorserver.sh` script stores the hashed value of *key* in the `connectorserver.key` property in the `ConnectorServer.properties` file.

10.3.2.5 Installing the Siebel Connector in the Java Connector Server

After you have installed the Java Connector Server, follow these steps to install the Siebel connector in the Java Connector Server:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the Siebel Connector.
2. Stop the Java Connector Server.
3. Copy the `org.identityconnectors.siebel-version.jar` file from the `CONNECTOR_INSTALL_DIR/WEB-INF/bundles` directory to the `CONNECTOR_SERVER_HOME/bundles` directory.
4. Depending on the target system you are using, copy the following Siebel JAR files into the `CONNECTOR_SERVER_HOME/lib` directory:
   - For Siebel 7.8 through 8.1.1: `Siebel.jar` and `SiebelJI_enu.jar`
   - For Siebel 7.5 through 7.7: `SiebelJI.jar`, `SiebelJI_Common.jar`, and `SiebelJI_enu.jar`

   The Siebel JAR files are available in the `SIEBEL_INSTALLATION_DIRECTORY/siebsrvr/CLASSES` directory.
5. Start the Java Connector Server.

10.3.3 Creating a Siebel Connector Resource

To create a Siebel connector resource, follow these steps:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the Siebel Connector.
2. Log in to the Oracle Waveset Administrator interface.
3. Create the Siebel connector resource by following the Create Siebel Connector Resource wizard.
   - Specify the appropriate version of the Siebel connector.
   - If you are installing the Siebel connector in the Java Connector Server, choose the desired Java Connector Server. However, if you are installing the Siebel connector in Oracle Waveset, specify the value for the Java Connector Server as Local.
4. Specify values for the configuration parameters, as described in Siebel Connector Configuration Parameters.

For additional information about creating resources, see "Understanding and Managing Waveset Resources" in the *Oracle Waveset 8.1.1 Business Administrator's Guide* in the following library:

http://docs.oracle.com/cd/E19225-01/index.html

10.4 Using the Siebel Connector

10.4.1 Siebel Connector Account Attributes

The default schema map assumes that the Employee business object and Employee business component are configured. You might have to add, remove, or change attributes to manage your Siebel environment, especially if you have configured the connector to use a business object or business component other than the default.

In the schema map, primary field values are handling by using a separate field that ends with `true`. For example, the `Employee;Position;Name;true` field sets the primary position for an employee. In the schema map, secondary field values are handling by using a separate field that ends with `false`. For example, the `Employee;Position;Name;false` filed sets the secondary position for an employee.

Table 10–3 Siebel Connector Account Attributes

Identity System User Attribute	Resource User Attribute	Description
`accountId`	`common;Login Name`	User's login name
`firstname`	`common;First Name`	User's first name
`lastname`	`common;Last Name`	User's last name
`middleName`	`common;Middle Name`	User's middle name
`Work Phone Extension`	`Work Phone Extension`	User's work phone number extension
`email`	`common;EMail Addr`	User's email address
`Alias`	`common;Alias`	User's alias
`Employee Type Code`	`Employee Type Code`	User's employee code
`Time Zone`	`common;Time Zone`	User's time zone name and translation
`Preferred Communications`	`common;Preferred Communications`	User's preferred method of communication
`Job Title`	`common;Job Title`	User's job title
`Home Phone`	`common;Home Phone #`	User's home phone number
`Personal Title`	`common;Personal Title`	User's personal title
`Phone`	`Employee;Phone #`	User's phone number
`Fax Number`	`common;Fax #`	User's FAX number
`UserType`	`UserType`	User's type. Value should be either `Employee` or `User`.

Table 10–3 (Cont.) Siebel Connector Account Attributes

Identity System User Attribute	Resource User Attribute	Description
Primary Responsibility	common;Responsibility;Name;true	Multi-value attribute that contains a list of responsibilities you want to assign to the employee. You must manage this attribute in the user form with the drop down box. All assigned responsibilities must exist in Siebel. To assign a Primary Responsibility, add the Primary Responsibility attribute to your schema map and set the attribute to the name of the responsibility you want to make primary.
Secondary Responsibility	common;Responsibility;Name;False	Multi-value attribute that contains a list of responsibilities you want to assign to the employee. You must manage this attribute in the user form with a multi-select box. The Responsibility field is set as a multi-select box in the sample Siebel User Form.
Primary Position	Employee;Position;Name;True	Multi-value attribute that contains a list of positions you want to assign to the employee. You must manage this attribute in the user form with the drop down box. All assigned positions must exist in Siebel. To assign a Primary Position, add the Primary Position attribute to your schema map and set the attribute to the name of the position you want to make primary.
Secondary Position	Employee;Position;Name;False	Multi-value attribute that contains a list of positions you want to assign to the employee. You must manage this attribute in the user form with a multi-select box. The Position field is set as a multi-select box in the sample Siebel User Form.

Note:
- A Resource User Attribute prefix with common; means that this attribute is used in both the User Business Object and Employee Business Object. So, it is common for both.
- A Resource User Attribute prefix with Employee; means that this attribute is specific to the Employee Business Object.
- A Resource User Attribute prefix with User; means that this attribute is specific to the User Business Object.

10.4.2 Siebel Connector Sample Form

The following sample form is provided with the Siebel connector:

Form	File
Siebel Connector User Form	sample/SiebelConnectorUserForm.xml

After you install the Siebel connector and import the Siebel Connector User Form, edit the Tabbed User Form, in order for values to be populated in the Position and Responsibility fields when a new user is created.

For example, to support the Siebel Connector user form, modify the Tabbed User Form, as follows:

1. Go to Oracle Waveset debug page:

   `http://host_name:port/idm/debug`

2. Select User Form from the drop-down box, which is adjacent to List Objects, and then click on List Objects.
3. Search for the Tabbed User Form and then click Edit.
4. Make the following changes in the Tabbed User Form:
   1. Add the `SiebelConnectorUserForm` inside the `<Include>` tag, as follows:

      ```
      <Include>
      ...
      <ObjectRef type='UserForm' name='SiebelConnectorUserForm'/>
      </Include>
      ```

   2. Add the following `<FormRef . . .>` element before the `<FormRef name='MissingFields'/>` tag:

      ```
      <FormRef name='SiebelConnectorUserForm'/>
      ```

For more information, see "Customizing Forms" in Chapter 2, Waveset Forms, in the *Oracle Waveset 8.1.1 Deployment Reference* (http://download.oracle.com/docs/cd/E19225-01/821-0378/bvaex/index.html).

10.4.3 Choosing Business Objects and Components

By default, the Siebel connector uses the User and Employee Siebel business component of the User and Employee Siebel business object for account provisioning.

You can use the Siebel Tools Client to inspect your business component and to verify which attributes are available for provisioning. The default schema map has some common attributes that are useful for the default Users and Employee business component.

You might have to add, remove, or change attributes to manage your Siebel environment, especially if you have configured the connector to use a business object or business component other than the default.

The following steps are a basic guide to discovering which attributes Oracle Waveset can provision to your Siebel environment using the Siebel Tools client.

To identify attributes for provisioning to a Siebel environment, follow these steps:

1. Open the Siebel Tools Object Explorer.
2. Click the **Business Component** icon.
3. Scroll down or create a query to select the desired business component.
4. Select **Fields** within the Object Explorer.

   A list of fields available to the business component is displayed.

   The field *Name* column values shown in the Object Explorer are typically used for the right-hand side (or the Resource User Attribute), within the schema map of your configured Siebel resource.

In general, you can manage any of these fields to some degree. For more information, see Siebel Connector Account Attributes.

10.4.4 Configuring the Siebel Connector for Multiple Versions of the Target System

To configure the Siebel connector for multiple versions of the Siebel target system, follow these steps:

1. Install the Java Connector Server, as described in Installing and Configuring the Java Connector Server.

   Note: If the third-party JAR files (this is, the Siebel JAR files) for each target system version are different, install a Java Connector Server for each target system you are using.

2. Deploy the Siebel connector in the Java Connector Server, as follows:

   1. Copy the `org.identityconnectors.siebel-`*`version`*`.jar` file from the `connector_install_dir/bundles` directory to the `CONNECTOR_SERVER_HOME/bundles` directory.

   2. Depending on the target system you are using, copy the following Siebel JAR files into the `CONNECTOR_SERVER_HOME/lib` directory:

      – For Siebel 7.8 through 8.1.1: `Siebel.jar` and `SiebelJI_enu.jar`

      – For Siebel 7.5 through 7.7: `SiebelJI.jar`, `SiebelJI_Common.jar`, and `SiebelJI_enu.jar`

      The Siebel JAR files are available in the `SIEBEL_INSTALLATION_DIRECTORY/siebsrvr/CLASSES` directory.

3. Start the Java Connector Server.

4. Create a new Resource instance for each additional version of the target system.

10.5 Troubleshooting the Siebel Connector

Use the Oracle Waveset debug pages to set trace options on the following connector class:

`org.identityconnectors.siebel`

This class returns the available error messages from the Siebel resource.

For more information, see Debugging and Troubleshooting.

11

Oracle Waveset Connector for SAP User Management Engine

This chapter includes the following information about the SAP User Management Engine (UME) connector for Oracle Waveset:

11.1 About the SAP UME Connector

11.1.1 Overview of the SAP UME Connector

The SAP UME connector for Oracle Waveset provides provisioning and reconciliation for SAP UME target systems. The connector uses the Service Provisioning Markup Language (SPML) to access these data sources on SAP UME target systems:

- System database of Application Server (AS) for Java (AS Java database)
- User Management of Application Server (AS) Advanced Business Application Programming (ABAP) database (AS ABAP database)
- Lightweight Directory Access Protocol (LDAP) directory service (LDAP directory service}

The SAP UME connector also supports remote role assignment in a Federated Portal Network (FPN) configuration. FPN allows organizations with multiple portals, both SAP and non-SAP, to share content between independent portals. In FPN, the producers hold and run the applications. The consumer manages the redirect to producer portals. Remote role assignment enables the consumer administrator to assign complete roles offered by an SAP producer.

The SAP UME connector is implemented using the Identity Connector Framework (ICF). The ICF provides a container that separates the connector bundle from the application. The ICF also provides common features that developers would otherwise need to implement on their own, such as connection pooling, buffering, time outs, and filtering. For more information about the ICF, see Chapter 1, "Identity Connectors Overview".

The SAP UME connector supersedes the SAP Enterprise Portal resource adapter. To migrate from a resource adapter deployment, see Migrating to the SAP UME Connector From a SAP Enterprise Portal Resource Adapter.

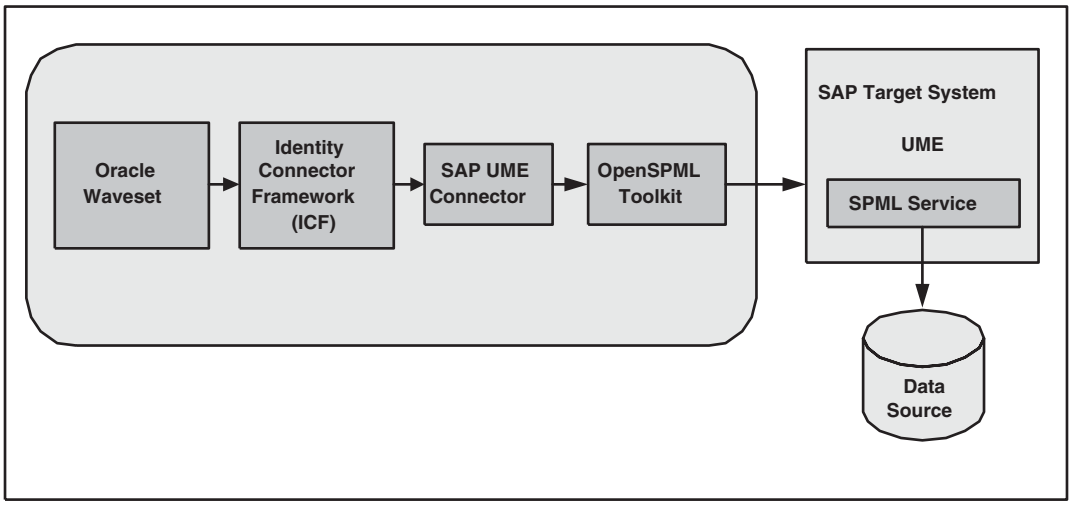
This section provides the following additional information about the SAP UME connector:

- SAP UME Connector Architecture
- SAP UME Connector Features
- Resource Configuration Parameters for the SAP UME Connector
- AS ABAP Data Source Constraints for the SAP UME Connector
- Limitations for UME Groups That Represent Roles in the AS ABAP
- Role Management With the SAP UME Connector

11.1.1.1 SAP UME Connector Architecture

The following figure shows the SAP UME connector architecture.

Figure 11–1 SAP UME Connector Architecture



The SAP UME connector architecture includes these components:

- Oracle Waveset includes the connector integration files. These files are XML files that provide the configuration information necessary to transform data from a resource to Oracle Waveset. Integration files are sometimes called the connector "glue" code.
- The Identity Connector Framework (ICF) provides basic provisioning, logging, and other functions that Oracle Waveset (and Oracle Identity Manager) connectors can use.

- The SAP UME connector uses the OpenSPML Toolkit to send requests to the SPML service running on the SAP UME target system.
- On the SAP UME target system, the SPML Service provides the provisioning and reconciliation capabilities for the specific data source. The SAP UME connector supports these data sources:
  - AS Java database
  - AS ABAP database
  - LDAP directory service

  The SAP UME connector supports agentless target deployment; that is, an agent is not required.

If you are installing the SAP UME connector in the Connector Server, see also SAP UME Connector Deployment Architecture With the Connector Server.

11.1.1.2 SAP UME Connector Features

The SAP UME connector for Oracle Waveset supports these operations:

Table 11–1 SAP UME Connector Operations

Operation	Description
Account provisioning	Operations include: ■ Create, modify, delete, lock, and unlock account ■ Enable and disable account The SAP UME does not have an explicit enable or disable account operation. However, you can enable or disable an account using the Valid Through user attribute on the SAP UME target system: ■ When a user is enabled from Oracle Waveset, the SAP UME connector will set the Valid Through attribute with the maximum date entered in the enable date field of the resource configuration. ■ When a user is disabled from Oracle Waveset, the SAP UME connector will set the Valid Through attribute with yesterday's date of Oracle Waveset. If that user has already logged in to the target system today or the password of that user was changed today, then the UME will update the Valid Through attribute with today's date and lock that user. The dates on Oracle Waveset and the SAP UME target system should be in sync. ■ Update account The SAP UME connector user form has the Valid Through attribute. You can enter any date value in this attribute field, which is similar to an update operation. ■ Change password ■ Add and remove role ■ Add and remove group In an AS ABAP database, only groups stored in the local database of AS Java can be assigned or unassigned. If groups stored as AS ABAP roles are selected, an exception error message is returned. If the SAP UME is configured with an ABAP data source, ABAP roles will be displayed as groups in the UME. Check whether the UME allows you to add a group that holds ABAP roles to a user. For example, a group from the `R3_ROLE_DS` data source. This scenario needs to be checked from the identity management screen of the UME. If the UME does not allow it, then the connector will throw an error if groups that represent ABAP roles are selected. In this scenario, to assign groups that represent an AS ABAP role, create a new AS Java role in the UME user administration tool and assign the group that represents the AS ABAP role to the new role. Assign this newly created AS Java role from Oracle Waveset.

Table 11–1 (Cont.) SAP UME Connector Operations

Operation	Description
Reconciliation	Only full and incremental reconciliation are supported. Active sync is not supported.

11.1.1.3 Resource Configuration Parameters for the SAP UME Connector

The SAP UME connector for Oracle Waveset supports the resource configuration parameters shown in the following table. These attributes must be defined in the SAP UME connector resource configuration (that is, `SAPUMEConfiguration.java`). The first column includes both the display name and field name for each parameter.

Table 11–2 Resource Configuration Parameters for the SAP UME Connector

Resource Configuration Parameter	Description
URL (`umeUrl`)	SPML service URL on the SAP target system, in the following format: `http(s)://`*`sap-target-system`*`:`*`port-number`*`/spml/spmlservice` For SSL communication, you must use the `https` protocol. For more information see Configuring SSL for the SAP UME Connector.
User ID (`umeUserId`)	User ID used for authentication.
Password (`umePassword`)	Password for the user ID used for authentication.
Enable Change Password (`changePwdFlag`)	Flag that specifies whether to change the password instead of resetting the password, in order to prevent users from changing the password at the first log on. Values can be Yes or No. The SAP UME target system expects the user to change the password at the next logon once the password is set by an administrator. To prevent this scenario, the first SPML request sets the password with a dummy password. Then, the second SPML request changes the password from the dummy password to the password entered in the process form. The security policy of some target systems allows changing the password only once per day. In this situation, the target system allows resetting the password and not changing the password. The target system will throw the error message "Could not update user NEW_PASSWORD_INVALID". In this situation, set this parameter value to No.
Dummy Password (`dummyPassword`)	Dummy password used if the `changePwdFlag` is set to Yes.
Enable Date (`enableDate`)	Maximum date to set while enabling a user, in the format: `yyyy/mm/dd`. The default value is `9999/12/31`.
Log SPML Request (`logSPMLRequest`)	To log SPML requests, set this parameter to Yes.

Table 11–2 (Cont.) Resource Configuration Parameters for the SAP UME Connector

Resource Configuration Parameter	Description
Password Handling Support (`pwdHandlingSupport`)	If the SAP UME is configured with an LDAP data source in writable mode, SSL configuration between the SAP UME and the LDAP data source is required for password management. With an LDAP data source configured in writable mode, if SSL is not configured between the SAP UME and the LDAP data source and the password need not be maintained from the SAP UME, set this parameter to No. Otherwise, set it to Yes.
Group Datasource (`groupDatasource`)	List of group data source names configured in the SAP UME. Default value is PRIVATE_DATASOURCE. The SAP UME does not allow adding a group from the Built-in Groups Adapter Data Source. Therefore, this data source should not be added in this configuration. To find a group data source name: 1. Login to the SAP UME administration console using the following URL: `http://host:port/useradmin` 2. In the identity management screen of the SAP UME, select Group in Search Criteria. 3. Select a data source and then click Go. For example: UME Database 4. Click any one group from the list. For example: Guests 5. Check Unique ID value of the field, using the format `GRUP.data-source-name.auto-generated-value`. For example: `GRUP.PRIVATE_DATASOURCE.un:Guests` 6. From this format, get the data source name. For example: PRIVATE_DATASOURCE Repeat these steps to get additional group data source names.

Table 11–2 (Cont.) Resource Configuration Parameters for the SAP UME Connector

Resource Configuration Parameter	Description
Role Datasource (`roleDatasource`)	List of role data source names configured in the SAP UME. Default value is UME_ROLE_PERSISTENCE. To find a role data source name: 1. Login to the SAP UME administration console using the following URL: `http://host:port/useradmin` 2. In the identity management screen of the SAP UME, select Role in Search Criteria. 3. Select a data source and then click Go. For example: UME Database 4. Click any one role from the list. For example: Administrator 5. Check Unique ID value of the field, using the format `ROLE.data-source-name.auto-generated-value`. For example: `ROLE.UME_ROLE_PERSISTENCE.un:Administrator` 6. From this format, get the data source name. For example: UME_ROLE_PERSISTENCE Repeat these steps to get additional role data source names.
LogonName Initial Substring (`logonNameInitialSubstring`)	Entry that specifies the set of characters allowed in the UME Logon Name. During reconciliation, the SAP UME connector gets users with the Logon Name attribute beginning with any of these characters. The default value is `"abcdefghijklmnopqrstuvwxyz1234567890"`. Any characters supported from other languages must be added to this entry. **Note**. This parameter provides a method to get users during reconciliation because the UME SPML API does not support getting the user records in a batch operation.

11.1.1.4 AS ABAP Data Source Constraints for the SAP UME Connector

An AS ABAP data source on the SAP UME target system has the following constraints for the SAP UME connector:

- Limitation when searching for users: The search considers only actions performed using the AS Java tools. Therefore, the SAP UME connector cannot search using the last modified timestamp.
- List of SAP UME user attributes: The list of user attributes that can be read from or written to the SAP UME with an AS ABAP data source is fixed and cannot be extended. However, a back-end AS ABAP system can have additional attributes, but these attributes are not supported from the SAP UME.
- Delay in the display of AS ABAP Roles in the SAP UME: If you create a new AS ABAP role or change the description of an existing AS ABAP role, these changes might not be visible in the SAP UME for up to 30 minutes. The SAP UME reads

this data from the AS ABAP every 30 minutes. To force the SAP UME to read the data from the AS ABAP, you must restart the AS Java. Therefore, SAP UME connector reconciliation might lose roles that have been recently created.

- Constraint in a Central User Administration (CUA) environment: The SAP UME can view only the roles that are present in the central system. Roles in child systems are not visible to the SAP UME. Therefore, you can view and maintain only role assignments to the central system from the SAP UME connector.
- The UME does not support maintaining the Form of Address and TimeZone attributes in an ABAP data source.

11.1.1.5 Limitations for UME Groups That Represent Roles in the AS ABAP

- You can assign ABAP users only to UME groups that represent ABAP roles.
- The UME cannot show a user-group assignment when the current date is outside the validity period of the corresponding user-role assignment in the AS ABAP.
- If you try to assign a UME group to a user when the user is already assigned to the corresponding ABAP role, but the current date is outside the validity period, you will receive an error message.
- If a role assignment to a user in ABAP is by means of a collective role or organizational management, you cannot unassign the user from the corresponding UME group.
- If a role assignment to a user in ABAP is by means of an indirect assignment through a reference user (visible in transaction SU01), you cannot unassign the user from the corresponding UME group.
- If a role assignment to a user in ABAP is by means of direct and indirect assignment simultaneously, you cannot unassign the user from the corresponding UME group.

  For example, a user administrator named `ADMIN` has assigned the user named `USER1` to the roles Z_DIRECT and Z_COLLECT. Z_COLLECT is a collective role including the role Z_DIRECT. When `ADMIN` uses identity management of the AS Java, `ADMIN` cannot unassign `USER1` from the UME group Z_DIRECT because this ABAP role is also assigned indirectly by the ABAP role Z_COLLECT.
- New groups created with the UME are stored in the local database.

11.1.1.6 Role Management With the SAP UME Connector

The SAP UME connector supports the assignment of the following types of roles to the user:

- Roles that defines what is displayed in the Portal
  - Portal roles - These roles are applicable to Enterprise Portal, and the connector supports the assignment of these roles to the user.
- Roles that defines what authorizations the user has in the backend system
  - UME authorization roles - These roles support programmatic authorization checks. The connector supports assignment of these roles.
  - J2EE Security role - These roles support declarative authorization checks. The connector does not support assignment of these roles. These roles need to be managed from the Visual Administrator tool of the J2EE Engine.
  - ABAP authorization role - These roles are applicable when UME is configured with an ABAP data source. These roles will be displayed as groups in the

UME. The UME instance needs to be checked whether it is supported or not. The connector will support the assignment of these roles if the UME instance supports it.

11.1.2 Security Considerations for the SAP UME Connector

- Supported Connections for the SAP UME Connector
- Required Administrative Privileges for the SAP UME Connector

11.1.2.1 Supported Connections for the SAP UME Connector

The supported connections for the SAP UME connector are:

- Following the SPML programming model, a URL connection must be created for the SAP Web AS SPML service using HTTP basic authentication.
- For secure communications between Oracle Waveset and the SAP Web Application Server, the SSL protocol (HTTPS) must be used. With SSL communications, data transferred is encrypted. For more information, see Configuring SSL for the SAP UME Connector.

11.1.2.2 Required Administrative Privileges for the SAP UME Connector

The user that connects to the SAP UME target system must have a security role assigned with the following UME actions:

- `UME.Spml_Read_Action`
- `UME.Spml_Write_Action`

If the SAP UME is configured as an AS ABAP data source and Central User Administration (CUA) is enabled in the back-end ABAP system, then a system must be assigned for this user. To assign a system, use transaction `SU01` from the back-end ABAP system.

11.1.3 Certified Components for the SAP UME Connector

The SAP UME connector for Oracle Waveset is certified with the following components:

Table 11–3 Certified Components for the SAP UME Connector

Component	Requirement
Oracle Waveset	Oracle Waveset 8.1.1 with Patch 6

Table 11–3 (Cont.) Certified Components for the SAP UME Connector

Component	Requirement
Target systems	■ SAP UME running on SAP NetWeaver '04 SPS 14 or later ■ SAP UME running on SAP NetWeaver 7.0 SPS 05 or later **Note**. When the SAP application is installed in the Java stack (such as Enterprise Portal), then the connector can connect to the UME of the SAP application. When the SAP application (such as SAP BW or SRM) is installed in the ABAP stack, SAP Enterprise Portal must be configured against the SAP application's (BW or SRM) user management. For this configuration, refer to the SAP target system documentation. The connector needs to connect to the UME of Enterprise Portal. When the SAP application (such as PI) is installed in a dual (ABAP plus Java) stack, then the connector can connect to the UME of the SAP application. However, all of the limitations for the ABAP data source will be applicable.
Identity Connector Framework (ICF)	ICF 1.0 or later
OpenSPML Toolkit	OpenSPML Toolkit v0.6 (included with the SAP UME connector bundle)
JDK or JRE	JDK or JRE 1.5 or later

11.1.4 Supported Languages for the SAP UME Connector

The SAP UME connector is localized in the following languages:

- Arabic
- Chinese (Simplified and Traditional)
- Czech
- Danish
- Dutch
- Finnish
- French
- German
- Greek
- Hebrew
- Hungarian
- Italian
- Japanese
- Korean
- Norwegian
- Polish
- Portuguese (Brazilian)

- Romanian
- Russian
- Slovak
- Spanish
- Swedish
- Thai
- Turkish

11.2 Migrating to the SAP UME Connector From a SAP Enterprise Portal Resource Adapter

If you currently have the SAP Enterprise Portal resource adapter installed, this section describes how to migrate the adapter to the SAP UME connector.

To migrate a SAP Enterprise Portal resource adapter, follow these steps:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the SAP UME Connector.
2. Log in to the Oracle Waveset Administrator interface.
3. Select the Resources tab and then the Migrate Adapters tab.
4. Follow the Migration Wizard to complete the migration. A script runs in the background that updates the schema map.

11.3 Deploying the SAP UME Connector

You can deploy the SAP UME connector either locally in Oracle Waveset or remotely in the Connector Server, as described in the following sections:

Note: In a production environment, it is recommended that you install the SAP UME connector in the Connector Server.

11.3.1 Installing the SAP UME Connector in the Connector Server

This section describes the following information about installing the SAP UME connector in the Connector Server:

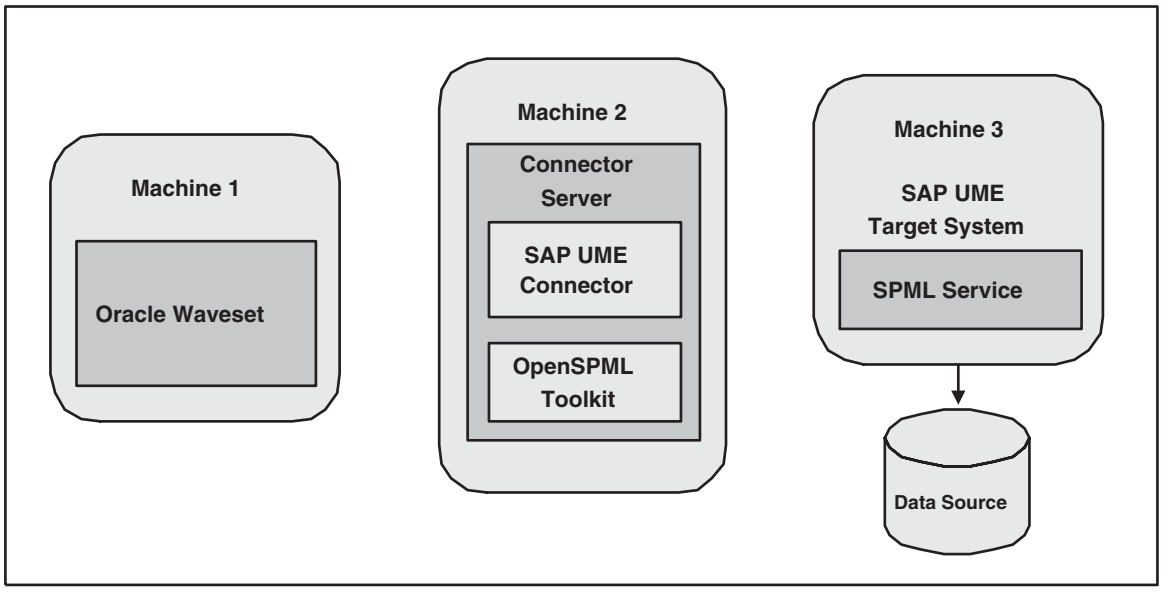
- SAP UME Connector Deployment Architecture With the Connector Server
- Installing and Configuring the Connector Server
- Running the Connector Server on Windows Systems
- Running the Connector Server on UNIX and Linux Systems
- Installing the SAP UME Connector in the Connector Server

Note: For the JDK requirements. see Certified Components for the SAP UME Connector. If necessary, see your `JAVA_HOME` environment variable to point to your specific installation.

11.3.1.1 SAP UME Connector Deployment Architecture With the Connector Server

If you install the SAP UME connector in the Connector Server, the following figure shows the distributed deployment architecture.

Figure 11–2 SAP UME Connector Deployment Architecture With the Connector Server



A SAP UME connector deployment with the Connector Server includes these components:

- **Machine 1** has Oracle Waveset deployed.
- **Machine 2** has the SAP UME connector installed in the Connector Server. The Connector Server is part of the Identity Connector Framework (ICF). The OpenSPML Toolkit is included with the SAP UME connector bundle.
- **Machine 3** has the SAP UME target system deployed. The SPML Service is used to access the Data Source.

11.3.1.2 Installing and Configuring the Connector Server

To install and configure the Connector Server, follow these steps:

1. Create a new directory on the machine where you want to install the Connector Server. In this section, `CONNECTOR_SERVER_HOME` represents this directory.
2. Unzip the Connector Server package in your new directory from Step 1. The Connector Server package is available with the Identity Connector Framework (ICF).
3. In the `ConnectorServer.properties` file, set the following properties, as required by your deployment. The `ConnectorServer.properties` file is located in the `conf` directory.

Property	Description
`connectorserver.port`	Port on which the Connector Server listens for requests. The default is 8759.
`connectorserver.bundleDir`	Directory where the connector bundles are deployed. The default is `bundles`.

Property	Description
connectorserver.libDir	Directory in which to place dependent libraries. The default is lib.
connectorserver.usessl	If set to true, the Connector Server uses SSL for secure communication. The default is false. If you specify true, use the following options on the command line when you start the Connector Server: ■ -Djavax.net.ssl.keyStore ■ -Djavax.net.ssl.keyStoreType (optional) ■ -Djavax.net.ssl.keyStorePassword See Bug 13343976: Connector Server With SSL is Not Working With the SAP UME Connector.
connectorserver.ifaddress	Bind address. To set this property, uncomment it in the file (if necessary). The bind address can be useful if there are more NICs installed on the machine.
connectorserver.key	Connector Server key.

4. Set the properties in the ConnectorServer.properties file, as follows:
   - To set connectorserver.key, run the Connector Server with the setKey option.
   - For all other properties, edit the ConnectorServer.properties file manually.
5. The conf directory also contains the logging.properties file, which you can edit if required by your deployment.

11.3.1.3 Running the Connector Server on Windows Systems

To run the Connector Server on Windows systems, use the ConnectorServer.bat script as follows:

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

   The ConnectorServer.bat script supports the following options:

Option	Description
/install [*serviceName*] ["-J *java option*"]	Installs the Connector Server as a Windows service. Optionally, you can specify a service name and Java options. If you do not specify a service name, the default name is ConnectorServerJava.
/run ["-J *java option*"]	Runs the Connector Server from the console. Optionally, you can specify Java options. For example, to run the Connector Server with SSL: ConnectorServer.bat /run "-J-Djavax.net.ssl.keyStore=mykeystore.jks" "-J-Djavax.net.ssl.keyStorePassword=*password*"

Option	Description
`/setkey [key]`	Sets the Connector Server key. The `ConnectorServer.bat` script stores the hashed value of the key in the `connectorserver.key` property in the `ConnectorServer.properties` file.
`/uninstall [serviceName]`	Uninstalls the Connector Server. If you do not specify a service name, the script uninstalls the `ConnectorServerJava` service.

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

11.3.1.4 Running the Connector Server on UNIX and Linux Systems

To run the Connector Server on UNIX and Linux systems, use the `connectorserver.sh` script, as follows:

1. Make sure that you have set the properties required by your deployment in the `ConnectorServer.properties` file, as described in Installing and Configuring the Connector Server.
2. Change to the `CONNECTOR_SERVER_HOME/bin` directory.
3. Use the `chmod` command to set the permissions to make the `connectorserver.sh` script executable.
4. Run the `connectorserver.sh` script. The script supports the following options:

Option	Description
`/run [ -Jjava-option ]`	Runs the Connector Server in the console. Optionally, you can specify one or more Java options. For example, to run the Connector Server with SSL: `./connectorserver.sh /run -J-Djavax.net.ssl.keyStore=mykeystore.jks -J-Djavax.net.ssl.keyStorePassword=password`
`/start [ -Jjava-option ]`	Runs the Connector Server in the background. Optionally, you can specify one or more Java options.
`/stop`	Stops the Connector Server, waiting up to 5 seconds for the process to end.
`/stop n`	Stops the Connector Server, waiting up to *n* seconds for the process to end.
`/stop -force`	Stops the Connector Server. Waits up to 5 seconds and then uses the `kill -KILL` command, if the process is still running.
`/stop n -force`	Stops the Connector Server. Waits up to *n* seconds and then uses the `kill -KILL` command, if the process is still running.
`/setKey key`	Sets the Connector Server key. The `connectorserver.sh` script stores the hashed value of *key* in the `connectorserver.key` property in the `ConnectorServer.properties` file.

11.3.1.5 Installing the SAP UME Connector in the Connector Server

To install the SAP UME connector for Oracle Waveset in the Connector Server, follow these steps:

1. Make sure you have installed Oracle Waveset with the patch shown in Certified Components for the SAP UME Connector.
2. Stop the Connector Server.
3. Copy the SAP UME connector bundle into the `CONNECTOR_SERVER_HOME/bundles` directory.
4. Start the Connector Server.

For information about starting and stopping the Connector Server, see Running the Connector Server on Windows Systems or Running the Connector Server on UNIX and Linux Systems.

11.4 Using the SAP UME Connector

- SAP UME Connector Account Attributes
- Sample Forms for the SAP UME Connector
- Configuring SSL for the SAP UME Connector

11.4.1 SAP UME Connector Account Attributes

The SAP UME connector supports the account attributes in the following object classes:

- User Object Class
- Group Object Class
- Role Object Class

11.4.1.1 User Object Class

All attributes are string data types. The first column includes both the display name and field name for each attribute.

Table 11–4 User Object Class Attributes

Attribute Name	Required	Description
Group Name (`assignedgroups`)	No	List of all directly assigned groups.
Role Name (`assignedroles`)	No	List of all directly assigned roles.
Data Source (`datasource`)	No	Home data source of the object.
Department (`department`)	No	Department code.
Display Name (`displayname`)	No	Display name.
Email (`email`)	No	Email address.
Fax (`fax`)	No	Complete FAX number.
First Name (`firstname`)	No	First name.
Unique ID (`id`)	No	Back-end ID.
Is user locked (`islocked`)	No	Specifies if the user is locked.
Job Title (`jobtitle`)	No	Job title.

Table 11–4 (Cont.) User Object Class Attributes

Attribute Name	Required	Description
Last Name (`lastname`)	Yes	Last name.
Language (`locale`)	No	Locale code.
Logon Name (`logonname`)	Yes	Unique name and logon ID. **Note:** The maximum Logon Name field length can vary on the SAP UME target system, depending on the specific data source configuration. For example, some target systems allow a maximum of 20 characters for the Logon Name field. If you specify a Logon Name field in Oracle Waveset that is greater than the number of characters allowed on the target system, the `LOGONID_TOO_LONG` error is returned. In this situation, specify a Logon Name field in Oracle Waveset less than or equal to the maximum number of characters allowed on the target system.
Mobile # (`mobile`)	No	Mobile phone number.
Salutation (`salutation`)	No	Salutation.
Security Policy (`securitypolicy`)	No	Type of security policy for the user (default, technical, or unknown). The default is `default`.
Telephone (`telephone`)	No	Complete telephone number.
Time Zone (`timezone`)	No	Time zone.
Title (`title`)	No	Title of the user.
Valid From (`validfrom`)	No	Date the user becomes valid.
Valid To (`validto`)	No	Date the user becomes invalid. Default is `9999-12-31`.
Street (`streetaddress`)	No	Home address of the user.
City (`city`)	No	Name of the city.
Zip code (`zip`)	No	Postal code of the city.
State (`state`)	No	Name of the state.
Country (`country`)	No	Country code following the ISO 3166 standard.

Note: By default, the SAP UME connector uses the password that is entered on the password reset page. To have a user wants set the password through the user form, add a new Identity System User Attribute on the account attribute page and map it to the native `password` attribute.

The schema details can change in different SAP NetWeaver releases. To support additional attributes, get the schema details from the `schema.xml` file that is provided with the AS Java.

11.4.1.2 Group Object Class

For the Group object class configuration, the `lstGroupDatasource` attribute in the user form specifies the name of the data source. The default names are `PRIVATE_DATASOURCE` and `SUPER_GROUPS_DATASOURCE`. All attributes are string data types. The first column includes both the display name and field name for each attribute.

Table 11–5 Group Object Class Attributes

Attribute Name	Required	Description
Unique Name (`uniquename`)	Yes	Name of the group.
Display Name (`displayname`)	No	Display name of the group.
Description (`description`)	No	Description of the group.
Group ID (`id`)	No	Back-end ID of the group.
User Members (`member`)	No	Assigned members of the group.
Assigned Roles (`assignedroles`)	No	List of all directly assigned roles.
Data Source (`datasource`)	No	Name of the data source.

Note: Maintaining a group in the connector using the SPML API is based on the data source configuration, as follows:

- If the UME is configured with an AS Java data source, groups are stored in the internal data source. Therefore, creating a new group using the SPML API is supported in an AS Java data source.
- If UME is configured with an AS ABAP data source, you can view AS ABAP roles as groups, but you cannot modify them or create new AS ABAP roles. You can create groups in the local AS Java database, which will not be reflected in the ABAP application. Therefore, creating a new role in an AS ABAP data source is not supported.

11.4.1.3 Role Object Class

For the Role object class configuration, the `lstRoleDatasource` attribute in the user form specifies the name of the data source. The default name is `UME_ROLE_PERSISTENCE`. All attributes are string data types. The first column includes both the display name and field name for each attribute.

Table 11–6 Role Object Class Attributes

Attribute Name	Required	Description
Unique Name (`uniquename`)	Yes	Name of the role.
Display Name (`displayname`)	No	Display name of the role.

Table 11–6 (Cont.) Role Object Class Attributes

Attribute Name	Required	Description
Description (`description`)	No	Description of the role.
User Members (`member`)	No	Assigned members of the role.
Role ID (`id`)	No	Back-end ID of the role.
Data Source (`datasource`)	No	Name of the data source.

Note: The SPML API does not support creating a new role. Creating an `AddRequest` in the `saprole` object class returns an `AddResponse` with the error as `UnsupportedOperation` with the message "Creation of new roles is not supported".

11.4.2 Sample Forms for the SAP UME Connector

The following sample forms are provided with the SAP UME connector:

Form	File
SAPUMEUserForm	`sample/SAPUMEUserForm.xml`
SAPUMEUserFormDefaultValues	`sample/SAPUMEUserFormDefaultValues.xml`
SAPUMEConnectorCreateGroupForm	`sample/SAPUMEConnectorCreateGroupForm.xml`
SAPUMEConnectorUpdateGroupForm	`sample/SAPUMEConnectorUpdateGroupForm.xml`
SAPUMEConnectorUpdateRoleForm	`sample/SAPUMEConnectorUpdateRoleForm.xml`

11.4.3 Configuring SSL for the SAP UME Connector

11.4.3.1 Configuring SSL for the SAP UME Connector With Oracle Waveset

This section describes how to configure Secure Sockets Layer (SSL) for the SAP UME connector installed in Oracle Waveset.

Before you configure SSL for the SAP UME connector, consider these requirements:

- On the SAP UME target system, SSL must be enabled for the specific data source. For information, see the documentation for your specific SAP UME target system.
- On the server where Oracle Waveset is deployed, a JDK or the JRE is required. For the requirements. see Certified Components for the SAP UME Connector. If necessary, set your `JAVA_HOME` or `JRE_HOME` environment variable to point to your specific installation.

To configure SSL for the SAP UME connector deployed in Oracle Waveset, follow these steps:

1. On the SAP UME target system, import the certificate from the data source into the SPML Provider:

a. Login to the SPML Provider on the SAP target system by specifying the following URL in your web browser:

   `https://sap-target-system:port-number/spml/spmlservice`

   The user name you specify to log in must have the following permissions: `UME.Spml_Read_Action` and `UME.Spml_Write_Action`.

b. Click Certificate Error, and then on the Certificate Invalid menu, click View Certificates.

c. Import the certificate from the SAP UME target system by clicking Install Certificate.

d. Specify the location for the certificate as the Certificate Store named Trusted Root Certificate Authorities.

2. While still logged into the SPLM Provider, select the certificate you imported in the previous step and export the certificate to a file:

   1. For the imported certificate, click the Details tab and then Copy to File... .
   2. For the Export File Format, check DER encoded binary X.509 (.CER).
   3. For the File to Export, enter the file name for the certificate. For example: `sapcert.cer`
   4. After you finish copying the certificate to a file, log out of the SPLM Provider.

3. On the application server you are using for Oracle Waveset, determine the certificate keystore location and import the certificate into the keystore. For example:

   ```
   keytool -import -alias cert-alias -keystore path-to-keystore-file -file
   certificate-file -storepass password
   ```

   where:

   - *cert-alias* is a user-defined alias for identification of the specific certificate in the certificate store.
   - *keystore-file* is the path to the keystore file.
   - *certificate-file* is the path to the certificate file you obtained from the SAP UME target system.
   - *password* is the password for the certificate store.

   For example, on a Windows system:

   ```
   keytool -import -alias sap-cert1
   -keystore C:\mydir\java\jre\lib\security\cacerts
   -file C:\mytagetcert\sapcert.cer -storepass changeit
   ```

   Note: The `keytool -importcert` command is supported in Java 6 or later releases. For Java 5 releases, use the `keytool -import` command.

   Make you have set your `JAVA_HOME` or `JRE_HOME` environment variable to point to your specific JDK or JRE installation

4. To verify the certificate in the certificate store, use the `keytool -list -keystore` command.

11.5 Troubleshooting the SAP UME Connector

Use the Oracle Waveset debug pages to set trace options on the following class:

`org.identityconnectors.sapume`

This class returns all available error messages from the SAP UME target resource.

11.6 Known Issues for the SAP UME Connector

11.6.1 Bug 13343976: Connector Server With SSL is Not Working With the SAP UME Connector

If you configure the SAP UME connector to communicate with the Connector Server using SSL, including setting the `connectorserver.usessl` property to `true` and importing the SAP UME target system certificate into the Connector Server JDK keystore, an attempt to access the SAP UME target system or run the Connector Server returns an error.

Workaround. None. Do not use SSL to communicate with the Connector Server.

Index

A

B

C

D

E

G

I

L

M

N

O

P

R

S

T

U

X