Oracle® Fusion Middleware

Administrator's Guide for Oracle Directory Integration Platform

11*g* Release 1 (11.1.1)

E10031-05

November 2011

ORACLE®

Oracle Fusion Middleware Administrator's Guide for Oracle Directory Integration Platform, 11*g* Release 1 (11.1.1)

E10031-05

Primary Author: Kevin Kessler

Contributing Author: Don Biasotti, Don Gosselin

Contents

3 Administering Oracle Directory Integration Platform

4 Managing the Oracle Directory Integration Platform

Part III Synchronization Using Oracle Directory Integration Platform

5 Understanding the Oracle Directory Synchronization Service

6 Configuring Directory Synchronization

7 Managing Directory Synchronization Profiles

8 Bootstrapping a Directory in Oracle Directory Integration Platform

9 Synchronizing with Tables in Oracle Database

10 Synchronizing with Oracle Human Resources

11 Synchronizing with Third-Party Metadirectory Solutions

16 Connected Directory Integration Concepts and Considerations

17 Configuring Synchronization with a Connected Directory

18 Integrating with Microsoft Active Directory

19 Deploying the Oracle Password Filter for Microsoft Active Directory

20 Integrating with Oracle Directory Server Enterprise Edition (Sun Java System Directory Server)

21 Integrating with IBM Tivoli Directory Server

22 Integrating with Novell eDirectory or OpenLDAP

23 Managing Integration with a Connected Directory

Part VI Appendixes

A Comparing Oracle Directory Integration Platform 11g Release 1 (11.1.1) and 10g Releases (10.1.4.x)

B Example Properties File for Synchronization Profiles

C Case Study: A Deployment of Oracle Directory Integration Platform

List of Figures

List of Tables

Preface

Oracle Fusion Middleware Administrator's Guide for Oracle Directory Integration Platform describes how to use Oracle Directory Integration Platform to integrate applications and directories—including third-party LDAP directories—with Oracle Internet Directory.

Audience

Oracle Fusion Middleware Administrator's Guide for Oracle Directory Integration Platform is intended for anyone who needs to integrate Oracle Internet Directory with applications and other directories—including third-party LDAP directories. You need to be familiar with either the UNIX/Linux operating systems or Microsoft Windows to understand the commands and examples in this guide.

To use this guide, you need some familiarity with the **Lightweight Directory Access Protocol (LDAP)**.

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.

Related Documentation

For more information, see:

- Online help available through Oracle Delegated Administration Services and Oracle Enterprise Manager.
- The Oracle Application Server, Oracle Database, and Oracle Identity Management documentation sets, especially:
  - *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*
  - *Oracle Fusion Middleware Getting Started with Oracle Identity Management*

- *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*
- *Oracle Fusion Middleware Guide to Delegated Administration for Oracle Identity Management*
- *Oracle Fusion Middleware Application Developer's Guide for Oracle Identity Management*
- *Oracle Enterprise Single Sign-On Suite Plus Administrator's Guide*
- *Oracle Application Server Certificate Authority Administrator's Guide*
- *Oracle Fusion Middleware Reference for Oracle Identity Management*
- *Oracle Fusion Middleware High Availability Guide*
- *Oracle Fusion Middleware Administrator's Guide*
- *Oracle Fusion Middleware Security Guide*
- *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*
- *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*

For additional information, see:

- Chadwick, David. *Understanding X.500—The Directory.* Thomson Computer Press, 1996.
- Howes, Tim and Mark Smith. *LDAP: Programming Directory-enabled Applications with Lightweight Directory Access Protocol.* Macmillan Technical Publishing, 1997.
- Howes, Tim, Mark Smith and Gordon Good, *Understanding and Deploying LDAP Directory Services.* Macmillan Technical Publishing, 1999.
- Internet Assigned Numbers Authority home page at `http://www.iana.org` for information about object identifiers.
- Internet Engineering Task Force (IETF) documentation available at: `http://www.ietf.org`, especially:
  - LDAPEXT charter and LDAP drafts
  - LDUP charter and drafts
  - RFC 2254, "The String Representation of LDAP Search Filters"
  - RFC 1823, "The LDAP Application Program Interface"
- The OpenLDAP Community at `http://www.openldap.org`

Conventions

The following text conventions are used in this document:

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

What's New in Oracle Directory Integration Platform?

The following is a list of new features in Oracle Directory Integration Platform 11*g* Release 1 (11.1.1):

- **Back-end Directory Support for Oracle Unified Directory and Oracle Directory Server Enterprise Edition**—As of 11*g* Release 1 (11.1.1.5) you have three directories to choose from when establishing a back-end directory. In addition to Oracle Internet Directory (OID), which previously was the only back-end directory option available, you can now choose Oracle Unified Directory (OUD) or Oracle Directory Server Enterprise Edition (ODSEE) as your Oracle Directory Integration Platform back-end directory. Both OUD and ODSEE support directory synchronization, but not provisioning. (Oracle Internet Directory (OID) still supports both directory synchronization and provisioning.) See Section 1.4.1, "Oracle Directory Integration Platform Back-End Directory" for additional information about the back-end directory.
- **Oracle Directory Integration Platform as a J2EE Application**—As of 11*g* Release 1 (11.1.1), Oracle Directory Integration Platform runs as a J2EE application. For more information, see "Oracle Directory Integration Platform" on page 1-5.
- **Integration with Fusion Middleware Control for Monitoring and Management**—In 11*g* Release 1 (11.1.1), you can use Oracle Enterprise Manager Fusion Middleware Control to monitor and manage Oracle Directory Integration Platform. For more information, see "Managing Oracle Directory Integration Platform Using Fusion Middleware Control" on page 4-6.
- **Updated Command Line Tools Using WebLogic Scripting Tools (WLST)**—Command line tools for Oracle Directory Integration Platform were updated in 11*g* Release 1 (11.1.1) to use the WebLogic Scripting Tool (WLST) framework. Although the command line tools use the WLST framework, you do not have to execute the commands using a `wlst` prefix. For more information, see "Command-Line Tools for Administering Oracle Directory Integration Platform" on page 3-3.
- **Custom Plug-in Support to Extend Data Transformation (Mapping)**—Oracle Directory Integration Platform 11*g* Release 1 (11.1.1) provides custom plug-in support to extend data transformation (mapping) functionality. This feature allows you to create and implement custom plug-ins for situations such as when you need to support new mapping operations and multiple literal values. For more information, see "Extending Mappings Using Custom Plug-ins" on page 6-15.
- **Domain and Attribute Exclusion in Map Files**—Oracle Directory Integration Platform 11*g* Release 1 (11.1.1) includes functionality that allows you to identify

domains and attributes to be excluded during bootstrap and synchronization. See "Excluding Domains" on page 6-6 and "Excluding Attributes" on page 6-8 for more information.

- **Custom Connector Support**—Oracle Directory Integration Platform 11*g* Release 1 (11.1.1) provides support for custom synchronization connectors. For more information on writing custom inbound and outbound connectors for Oracle Directory Integration Platform, see "Writing Custom Synchronization Connectors" on page 17-20.
- **Integration with Fusion Middleware Infrastructure (Audit, Logging, Credential Store Framework)**—Oracle Directory Integration Platform 11*g* Release 1 (11.1.1) utilizes the Oracle Fusion Middleware infrastructure for auditing, logging and credential storing. For more information, see:
  - "Managing Oracle Directory Integration Platform Logging Using Fusion Middleware Control" on page 4-8
  - "Auditing Oracle Directory Integration Platform Using Fusion Middleware Control" on page 4-8
  - "Credential Storing" on page 2-5
- **New Title for this Document**: For 11g Release 1 (11.1.1.2.0), this document was renamed from *Oracle Fusion Middleware Integration Guide for Oracle Identity Management* to its current title of *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Integration Platform*.

Part I

Getting Started with Oracle Directory Integration Platform

This part discusses the concepts, components, architecture, and security features of Oracle Directory Integration Platform. It contains these chapters:

1

Introduction to Oracle Identity Management Integration

This chapter introduces Oracle Identity Management integration, its components, structure, and administration tools.

This chapter contains these topics:

- Why Oracle Identity Management Integration?
- Oracle Identity Management Installation Options
- Synchronization, Provisioning, and the Differences Between Them
- Components Involved in Oracle Identity Management Integration

> **See Also:** Appendix C, "Case Study: A Deployment of Oracle Directory Integration Platform" for an example of how you can deploy Oracle Identity Management integration

1.1 Why Oracle Identity Management Integration?

Oracle Directory Integration Platform enables you to reduce administrative time and costs by integrating your applications and directories—including third-party LDAP directories—with a master back-end directory. Oracle Directory Integration Platform (DIP) supports the use of either Oracle Internet Directory, Oracle Unified Directory, or Oracle Directory Server Enterprise Edition as a back-end directory.

Use Oracle Directory Integration Platform to achieve these example objectives:

- Keep employee records in Oracle Human Resources consistent with those in the Oracle back-end directory. Oracle Directory Integration Platform provides this synchronization through the Oracle Directory Synchronization Service.
- Notify certain LDAP-enabled applications—such as Oracle Application Server Portal (Oracle Portal)—whenever changes are applied to the Oracle back-end directory. The Oracle Directory Integration Platform provides this notification through its Oracle Directory Integration Platform Provisioning Service.

Throughout the integration process, Oracle Directory Integration Platform ensures that the applications and other directories receive and provide the necessary information in a reliable way.

You can integrate with various directories, including the following:

- Microsoft Active Directory 2003 and 2008
- Active Directory Application Mode or ADAM, version 1 with SP1 on Windows 2003 (Microsoft Active Directory Lightweight Directory Service (AD LDS))

- Oracle Directory Server Enterprise Edition (ODSEE) 11gR1 (11.1.1.3+), which was previously Sun Java System Directory Server
- Novell eDirectory 8.8
- OpenLDAP 2.2
- IBM Tivoli Directory Server 6.2
- Oracle Unified Directory 11gR1 (11.1.1.5+)

See Also: For a complete list of supported directories, refer to the *System Requirements and Supported Platforms for Oracle Fusion Middleware 11gR1* certification matrix, available here:

`http://www.oracle.com/technetwork/middleware/downloads/fmw-11gr1certmatrix.xls`

Click the "FMW on WLS - Id&Access" tab and refer the "Directory Integration Platform (DIP)" row that appears under the heading "LDAP Certifications for Oracle Fusion Middleware 11gRelease1 (11.1.1.x)."

For example, in an Oracle Fusion Middleware environment, where access to Oracle components relies on data stored in an Oracle directory, you can still use Microsoft Active Directory as the central enterprise directory. Users of that directory can still access Oracle components because Oracle Directory Integration Platform can synchronize the data in Microsoft Active Directory with that in Oracle Internet Directory, Oracle Unified Directory, or Oracle Directory Server Enterprise Edition.

See Also:

- Chapter 10, "Synchronizing with Oracle Human Resources"
- Chapter 18, "Integrating with Microsoft Active Directory"
- Chapter 20, "Integrating with Oracle Directory Server Enterprise Edition (Sun Java System Directory Server)"

1.2 Oracle Identity Management Installation Options

Oracle Directory Integration Platform can be installed simultaneously with other Oracle Identity Management components on the same host (server), or by itself as a standalone instance on a host system separate from other Oracle Identity Management components. This could be the case if you want to separately manage J2EE based components (like DIP, ODSM, or FMW control) in a dedicated WebLogic domain on a dedicated server.

To install a standalone Oracle Directory Integration Platform instance, you first need to install an Oracle Internet Directory or Oracle Unified Directory or Oracle Directory Server Enterprise Edition component. You should install a standalone instance of Oracle Directory Integration Platform under the following circumstances:

- You need Oracle Directory Integration Platform to be installed in a different application server instance.
- The applications that you need to provision and synchronize require intensive processing.
- You need to run multiple instances of Oracle Directory Integration Platform for high availability.

See: The *Oracle Fusion Middleware Installation Guide for Oracle Identity Management* for complete information about installing Oracle Directory Integration Platform.

1.3 Synchronization, Provisioning, and the Differences Between Them

Synchronization has to do with directories rather than applications. It ensures the consistency of entries and attributes that are in both the Oracle back-end directory and the other connected directories.

Note: Synchronization and Replication *are not* synonymous. Replication is used for data handling between directories of the same vendor. Synchronization, on the other hand, provides better control of data that has to be kept synchronized between the backend directory (metadirectory) and all connected third-party directories based on the transformation and mapping rules DIP provides.

Provisioning has to do with applications. It notifies them of changes to user or group entries or attributes that the application needs to track.

This section contains these topics:

- Synchronization
- Provisioning
- How Synchronization and Provisioning Differ

1.3.1 Synchronization

Synchronization enables you to coordinate changes between the Oracle back-end directory and the connected directories. To ensure that all directories use and provide only the latest data, each directory must be informed of changes made in the other connected directories. Synchronization ensures that changes to directory information—including, but not limited to data updated through provisioning—is kept consistent.

A single Directory Integration Platform service can simultaneously handle synchronization duties between multiple connected directories and the Oracle back-end directory. To connect an additional directory to the Oracle back-end directory, create a synchronization profile for that specific directory. This profile specifies the format and content of the data to be synchronized between the Oracle back-end directory and the connected directory. To create a synchronization profile, you can use the `manageSyncProfiles` utility or Oracle Enterprise Manager Fusion Middleware Control.

See Also: Part III, "Synchronization Using Oracle Directory Integration Platform"

1.3.2 Provisioning

Provisioning enables you to ensure that an application is notified of directory changes to, for example, user or group information. Such changes can affect whether the application allows a user access to its processes and determines which resources can be used.

Use provisioning when you are designing or installing an application has the following requirements:

- Does not maintain a directory
- Is LDAP-enabled
- Can and should allow only authorized users to access its resources

When you install an application that you want to provision, you must create a provisioning integration profile for it by using the `oidprovtool` utility.

> **See Also:** Part IV, "Provisioning with the Oracle Directory Integration Platform"

1.3.3 How Synchronization and Provisioning Differ

Synchronization and provisioning have important operational differences, as described in Table 1–1.

Table 1–1 Directory Synchronization and Provisioning Integration Distinctions

Consideration	Directory Synchronization	Provisioning Integration
The time for action	Application deployment time. Directory synchronization is for connected directories requiring synchronization with the Oracle back-end directory.	Application design time. Provisioning integration is for application designers developing LDAP-enabled applications.
Communication direction	Either one-way or two-way—that is, either from the Oracle back-end directory to the connected directories (including one or more connected Oracle databases), the reverse, or both.	Either one-way or two-way—that is, either from the Oracle back-end directory to applications, the reverse, or both.
Type of data	Any data in a directory.	Restricted to provisioned users and groups.
Examples	Oracle Human Resources Oracle Directory Server Enterprise Edition (Sun Java System Directory Server) Oracle Unified Directory Oracle Internet Directory Microsoft Active Directory Novell eDirectory OpenLDAP IBM Tivoli Directory Server Oracle Database	Oracle Portal

1.4 Components Involved in Oracle Identity Management Integration

This section describes the components involved in Oracle Identity Management integration. It contains these topics:

- Oracle Directory Integration Platform Back-End Directory
- Oracle Directory Integration Platform

- Oracle Application Server Single Sign-On

1.4.1 Oracle Directory Integration Platform Back-End Directory

Either Oracle Internet Directory, Oracle Unified Directory, or Oracle Directory Server Enterprise Edition can be used as the repository in which Oracle components and third-party applications store and access user identities and credentials. The Oracle back-end directory uses the Oracle directory server to authenticate users by comparing the credentials entered by users with the credentials stored in the Oracle directory.

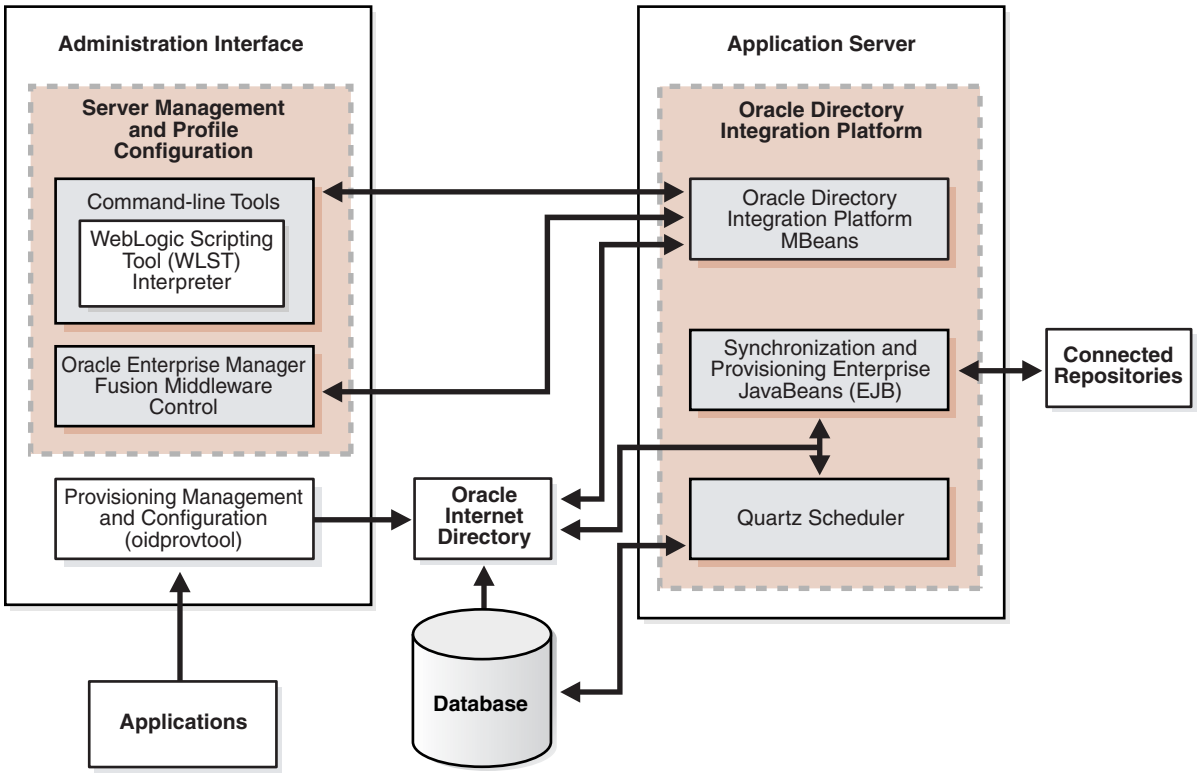
When credentials are stored in a connected directory and not in the Oracle back-end directory, users can still be authenticated if you are using Oracle Internet Directory as the back-end directory. In this case, Oracle Internet Directory uses an external authentication plug-in that authenticates users against the connected directory server. Currently, the external authentication plug-in *is not* available for Oracle Unified Directory or Oracle Directory Server Enterprise Edition, so if either of these directories are utilized as your back-end directory, users cannot be authenticated if credentials are stored in a connected directory.

1.4.2 Oracle Directory Integration Platform

The Oracle Directory Integration Platform is a J2EE application that enables you to synchronize data between different repositories and the Oracle back-end directory. Oracle Directory Integration Platform includes services and interfaces that allow you to develop synchronization solutions with other enterprise repositories. It can also provide interoperability between third party metadirectory solutions and Oracle directories.

Figure 1–1 shows an example of an Oracle Directory Integration Platform environment:

Figure 1–1 Example of an Oracle Directory Integration Platform Environment



In the example in Figure 1–1, the Oracle back-end directory is synchronized with connected directories using Oracle Directory Integration Platform's Synchronization Enterprise JavaBeans (EJB) and the Quartz Scheduler. Similarly, changes in the Oracle back-end directory are sent to various repositories using Oracle Directory Integration Platform's Provisioning Enterprise JavaBeans (EJB) and the Quartz Scheduler.

1.4.2.1 Understanding the Oracle Directory Integration Platform Server

The Oracle Directory Integration Platform Server performs the following services:

- Oracle Directory Integration Platform Synchronization Service:
  - Scheduling—Processing a synchronization profile based on a predefined schedule
  - Mapping—Executing rules for converting data between connected directories and the Oracle back-end directory
  - Data propagation—Exchanging data with connected directories by using a connector
  - Error handling
- Oracle Directory Integration Platform Provisioning Service:
  - Scheduling—Processing a provisioning profile based on a predefined schedule
  - Event Notification—Notifying an application of a relevant change to the user or group data stored in the Oracle back-end directory
  - Error handling

See Also: Chapter 4, "Managing the Oracle Directory Integration Platform"

1.4.2.2 Understanding the Oracle Directory Integration Platform Synchronization Service

In the Oracle Directory Integration Platform environment, the contents of connected directories are synchronized with the Oracle back-end directory through the Oracle Directory Integration Platform Synchronization Service, which includes Synchronization Enterprise JavaBeans (EJB) and the Quartz Scheduler.

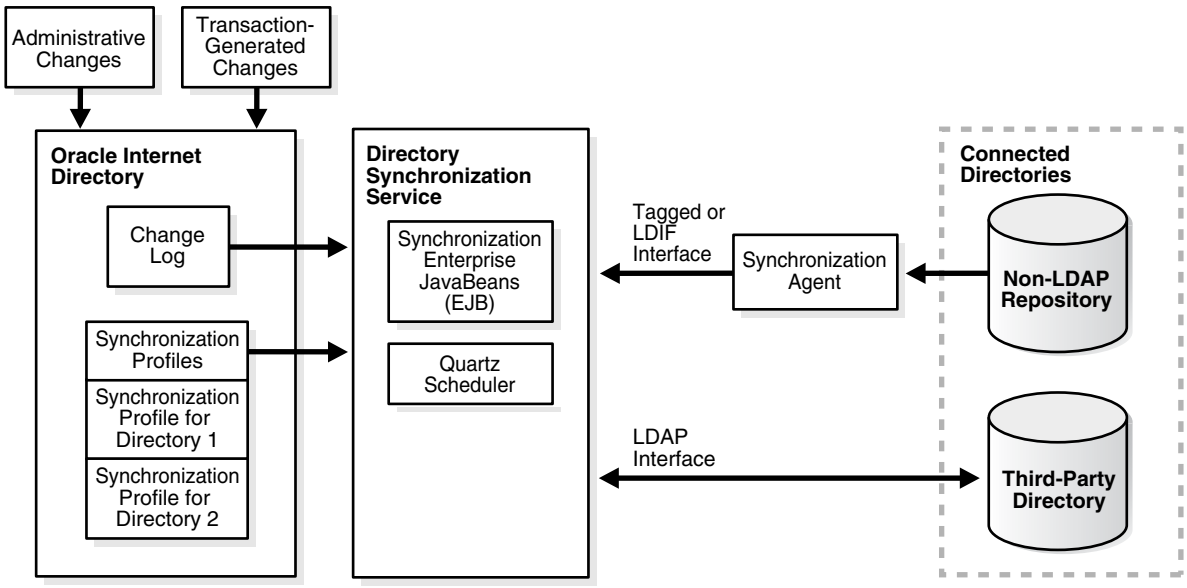
For Oracle Fusion Middleware components, the Oracle back-end directory is the central directory for all information, and all other directories are synchronized with it. This synchronization can be:

- One-way: Some connected directories only supply changes to the Oracle back-end directory and do not receive changes from it. This is the case, for example, with Oracle Human Resources, the primary repository and basis for comparison for employee information.
- Two-way: Changes in the Oracle back-end directory can be exported to connected directories, and changes in connected directories can be imported into the Oracle back-end directory.

Certain attributes can be targeted or ignored by the synchronization service. For example, the attribute for the employee badge number in Oracle Human Resources may not be of interest to the Oracle back-end directory, its connected directories, or client applications. You might not want to synchronize them. On the other hand, the employee identification number may be of interest to those components, so you might want to synchronize them.

Figure 1–2 shows the interactions among components in the Oracle Directory Synchronization Service in a sample deployment.

Figure 1–2 Interactions of the Oracle Directory Integration Platform Synchronization Service



The central mechanism triggering all such synchronization activities is the Oracle back-end directory change log. It adds one or more entries for every change to any

connected directory, including the Oracle back-end directory. The Oracle Directory Synchronization Service:

- Monitors the change log.
- Takes action whenever a change corresponds to one or more synchronization profiles.
- Supplies the appropriate change to all other connected directories whose individual profiles correspond to the logged change. Such directories could include, for example, Oracle RDBMS, Oracle Human Resources, Microsoft Active Directory, Oracle Unified Directory, Oracle Directory Server Enterprise Edition (Sun Java System Directory Server), Novell eDirectory, IBM Tivoli Directory Server, or OpenLDAP. The Oracle Directory Synchronization Service supplies these changes using the interface and format required by the connected directory. Synchronization through the Oracle Directory Integration Platform connectors ensures that the Oracle back-end directory remains up-to-date with all the information that the Oracle back-end directory clients need.

1.4.2.3 Understanding the Oracle Directory Integration Platform Provisioning Service

The Oracle Directory Integration Platform Provisioning Service, which includes Provisioning Enterprise JavaBeans (EJB) and the Quartz Scheduler, ensures that each provisioned application is notified of changes in, for example, user or group information. To do this, it relies on the information contained in a provisioning integration profile. Each provisioning profile:

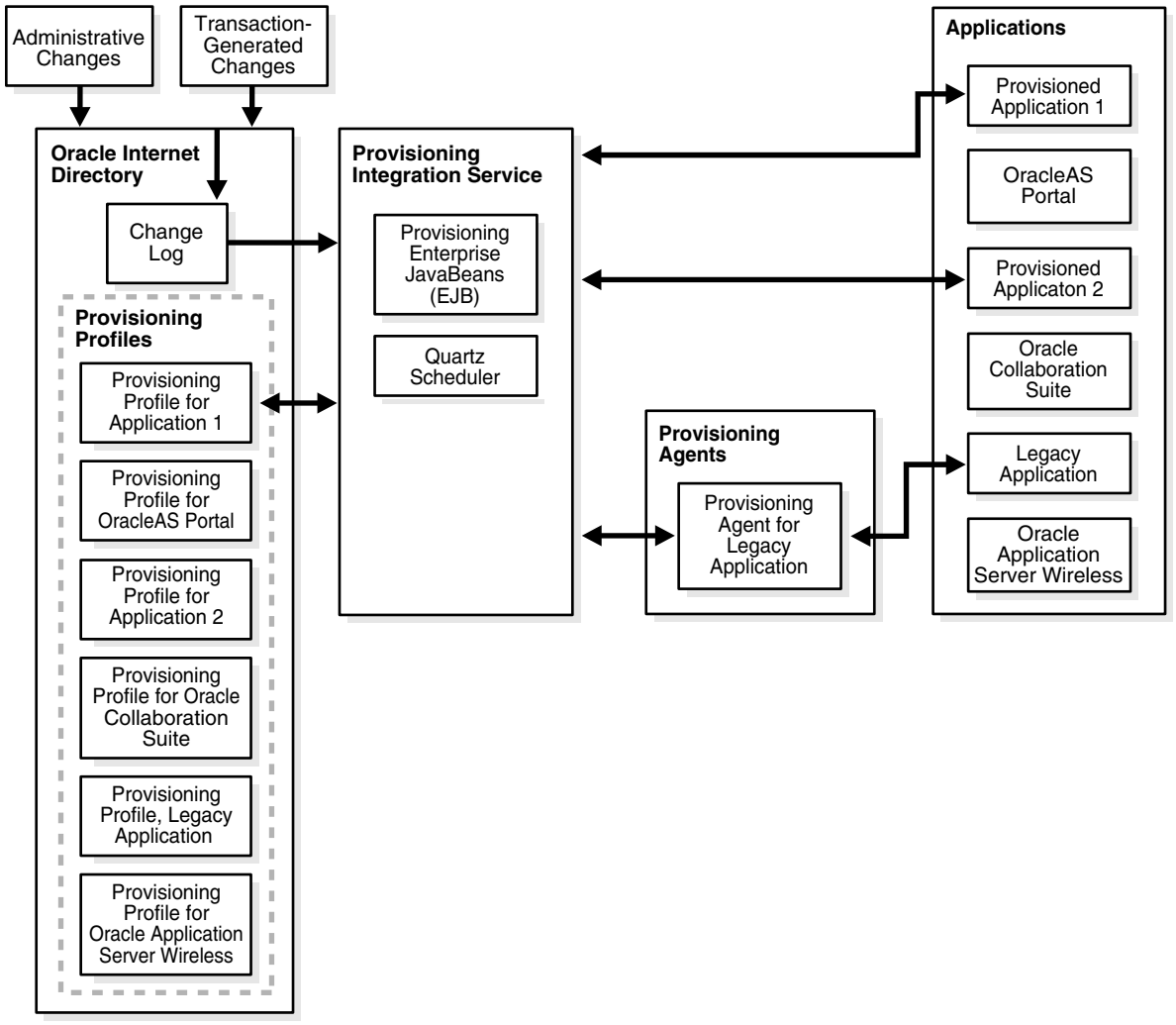
- Uniquely identifies the application and organization to which it applies
- Specifies, for example, the users, groups, and operations requiring the application to be notified

When changes in the Oracle back-end directory match what is specified in the provisioning profile of an application, the Oracle Directory Integration Platform Service sends the relevant data to that application.

> **Note:** A legacy application—that is, one that was operational before the Oracle Directory Integration Platform Service was installed—would not have subscribed in the usual way during installation. To enable such an application to receive provisioning information, a **provisioning agent**, in addition to the provisioning profile, must be developed. The agent must be able to translate the relevant data from the Oracle back-end directory into the exact format required by the legacy application.

Figure 1–3 shows the interactions among components in an Oracle Directory Integration Platform Service environment, including the special case of a provisioning agent for a legacy application.

Figure 1–3 Interactions of the Oracle Directory Integration Platform Provisioning Service



1.4.3 Oracle Application Server Single Sign-On

Oracle Application Server Single Sign-On (OracleAS Single Sign-On Server) enables users to access Oracle Web-based components by logging in only once.

Oracle components delegate the login function to the OracleAS Single Sign-On Server. When a user first logs in to an Oracle component, the component redirects the login to the OracleAS Single Sign-On Server. The OracleAS Single Sign-On Server authenticates the user by verifying the credentials entered by the user against those stored in the Oracle back-end directory. After authenticating the user, and throughout the rest of the session, the OracleAS Single Sign-On Server grants the user access to all the components the user both seeks and is authorized to use.

Note: Oracle Directory Integration Platform 11*g* Release 1 (11.1.1) interoperates with and supports Oracle Application Server Single Sign-On 10*g* Release 10.1.4.3.0.

Be aware that Oracle Application Server Single Sign-On has been deprecated in favor of Oracle Access Management 11gR1 single sign-on. Customers are advised to migrate to Oracle Access Management 11gR1 as soon as it is certified with their products.

For more information see the Oracle Lifetime Support Policy for Oracle Single Sign-on:

http://www.oracle.com/us/support/lifetime-support/index.html

See Also: *Oracle Enterprise Single Sign-On Suite Plus Administrator's Guide* for information about OracleAS Single Sign-On Server

2

Security Features in Oracle Directory Integration Platform

This chapter discusses the most important aspects of security in Oracle Directory Integration Platform. It contains these topics:

- Authentication in Oracle Directory Integration Platform
- Access Control and Authorization and Oracle Directory Integration Platform
- Data Integrity and Oracle Directory Integration Platform
- Data Privacy and Oracle Directory Integration Platform
- Tools Security and Oracle Directory Integration Platform
- Credential Storing

2.1 Authentication in Oracle Directory Integration Platform

Authentication is the process by which the Oracle directory server establishes the true identity of the user connecting to the directory. It occurs when an LDAP session is established by means of the `ldapbind` operation.

It is important that each component in Oracle Directory Integration Platform be properly authenticated before it is allowed access to the directory.

This section contains these topics:

- Secure Sockets Layer and Oracle Directory Integration Platform
- Oracle Directory Integration Platform Authentication in SSL Mode
- Profile Authentication

2.1.1 Secure Sockets Layer and Oracle Directory Integration Platform

The Oracle back-end directory should be configured to use **Secure Socket Layer (SSL)**.

If Oracle Unified Directory or Oracle Directory Server Enterprise Edition is your Oracle back-end directory, Oracle Directory Integration Platform will work in non-SSL mode when it is first installed. If Oracle Internet Directory is your Oracle back-end directory, however, an SSL connection is required.

Oracle Directory Integration Platform supports these SSL implementation modes:

- No Authentication (SSL Mode 1)—Provides SSL data encryption, but does not use SSL for authentication.

> **Note:** Oracle Directory Integration Platform only supports the No Authentication SSL mode (SSL mode 1) if your Oracle back-end directory is Oracle Internet Directory. If Oracle Unified Directory or Oracle Directory Server Enterprise Edition is your Oracle back-end directory, SSL Server Authentication (SSL mode 2) is your only SSL option.

- SSL Server Authentication (SSL Mode 2)—Includes both SSL data encryption and SSL authentication of the server to the client. In Oracle Directory Integration Platform, the server is the directory server, and the client is the Oracle Directory Integration Platform.

  The server verifies its identity to the client by sending a **certificate** issued by a trusted **certificate authority (CA)**. This mode requires a public key infrastructure (PKI) and certificates to be stored in the Java Keystore (JKS).

To use SSL with Oracle Directory Integration Platform, you must start both the Oracle back-end directory and Oracle Directory Integration Platform in the same SSL mode. For example, if the Oracle back-end directory is running in SSL mode 1, then Directory Integration Platform must be configured to connect to the Oracle back-end directory using the same SSL mode 1.

> **See Also:** If using Oracle Internet Directory as the Oracle back-end directory, refer to the chapter on preliminary tasks and information in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for instructions about starting the Oracle directory server in SSL mode.

2.1.2 Oracle Directory Integration Platform Authentication in SSL Mode

The identity of the directory server can be established by starting both the Oracle back-end directory and the directory integration server in SSL server authentication mode. In this case, the directory server provides its certificate to the directory integration server, which acts as the client of the Oracle back-end directory.

You can also configure the Oracle Directory Integration Platform to use SSL when connecting to a third-party directory. In this case, you store the connected directory certificates in the Java Keystore (JKS) as described in "Managing the SSL Certificates of Back-End Directories and Connected Directories" on page 4-14.

2.1.3 Profile Authentication

Within the Oracle back-end directory, an integration profile represents a user with its own distinguished name (DN) and password.

The users who can access the profiles are:

- The administrator of Oracle Directory Integration Platform.

  In Oracle Internet Directory the administrator is represented by the DN `cn=dipadmin,cn=dipadmins,cn=directory integration platform,cn=products,cn=oraclecontext`.

  In Oracle Unified Directory and Oracle Directory Server Enterprise Edition the administrator is represented by the DN `cn=dipadmin,cn=dipadmins,cn=directory integration platform,<suffix>`.

- Members of the Oracle Directory Integration Platform administrator group.

  In Oracle Internet Directory the administrator group is represented by the DN `cn=odisgroup,cn=DIPadmins,cn=Directory Integration Platform,cn=Products,cn=OracleContext`.

  In Oracle Unified Directory and Oracle Directory Server Enterprise Edition the administrator is represented by the DN `cn=odisgroup,cn=DIPadmins,cn=Directory Integration Platform,<suffix>`.

When the Oracle Directory Integration Platform imports data to the Oracle back-end directory based on an integration profile, it proxy-binds to the directory as that integration profile. The Oracle Directory Integration Platform can bind in either SSL or non-SSL mode.

2.2 Access Control and Authorization and Oracle Directory Integration Platform

Authorization is the process of ensuring that a user reads or updates only the information for which he or she has privileges. When directory operations are attempted within a directory session, the directory server ensures that the user—identified by the authorization identifier associated with the session—has the requisite permissions to perform those operations. If the user does not have the necessary permissions, then the directory server disallows the operation. Through this mechanism, called access control, the directory server protects directory data from unauthorized operations by directory users.

To restrict access to only the desired subset of Oracle Internet Directory data, for both the directory integration server and a connector, place appropriate access policies in the directory.

This section discusses these policies in detail. It contains these topics:

- Access Controls for the Oracle Directory Integration Platform
- Access Controls for Profiles

2.2.1 Access Controls for the Oracle Directory Integration Platform

The Oracle Directory Integration Platform binds to the directory both as itself and on behalf of the profile, as follows:

- When it binds as itself, it can cache the information in various integration profiles. This enables the directory integration server to schedule synchronization actions to be carried out by various connectors.
- When the directory integration server operates on behalf of a profile, it acts as proxy for the profile—that is, it uses the profile credentials to bind to the directory and perform various operations. The directory integration server can perform only those operations in the directory that are permitted in the profile.

To establish and manage access rights granted to directory integration servers, Oracle Directory Integration Platform creates a group entry, called `odisgroup`, during installation. When a directory integration server is registered, it becomes a member of this group.

In Oracle Internet Directory the DN of `odisgroup` is:

```
cn=odisgroup,cn=directory admins,cn=directory integration
plataform,cn=products,cn=oraclecontext
```

In Oracle Unified Directory and Oracle Directory Server Enterprise Edition the DN of `odisgroup` is:

```
cn=odisgroup,cn=directory admins,cn=directory integration
plataform,<suffix>
```

You control the access rights granted to directory integration servers by placing access control policies in the `odisgroup` entry. The default policy grants various rights to directory integration servers for accessing the profiles. For example, the default policy enables the directory integration server to compare user passwords between the Oracle back-end directory and the connected directory it binds as a proxy on behalf of a profile. It also enables directory integration servers to modify status information in the profile—such as the last successful execution time and the synchronization status.

2.2.2 Access Controls for Profiles

During installation, Oracle Directory Integration Platform creates a group entry called `odipgroup` that enables you to control the access rights granted to various profiles. For additional security, the `odipigroup` and `odipegroup` groups are also created during installation. All import profiles are assigned to the `odipigroup` group and all export profiles are assigned to the `odipegroup` group. Rights are controlled by placing appropriate access policies in the `odipgroup` entry. The default access policy, automatically installed with the product, grants to profiles certain standard access rights for the integration profiles they own. One such right is the ability to modify status information in the integration profile, such as the parameter named `orclodipConDirLastAppliedChgTime`. The default access policy also permits profiles to access Oracle Internet Directory change logs, to which access is otherwise restricted.

> **See Also:** The chapter on access control, specifically, the section about security groups, in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for instructions about setting access control policies for group entries

2.3 Data Integrity and Oracle Directory Integration Platform

Oracle Directory Integration Platform ensures that data is not modified, deleted, or replayed during transmission by using SSL. This SSL feature generates a cryptographically secure message digest—through cryptographic checksums using either the Message-Digest algorithm 5 (MD5) or the Secure Hash Algorithm (SHA)—and includes the message digest with each packet sent across the network.

2.4 Data Privacy and Oracle Directory Integration Platform

Oracle Directory Integration Platform ensures that data is not disclosed during transmission by using public-key encryption available with SSL. In public-key encryption, the sender of a message encrypts the message with the public key of the recipient. Upon delivery, the recipient decrypts the message using the recipient's private key.

To exchange data securely between the directory integration server and Oracle Internet Directory, you must run both components in the same SSL mode.

2.5 Tools Security and Oracle Directory Integration Platform

You can run all the commonly used tools in SSL mode to transmit data to Oracle Internet Directory securely, including Oracle Enterprise Manager Fusion Middleware Control.

2.6 Credential Storing

Oracle Directory Integration Platform uses the Credential Store Framework of the Oracle Application Server 11g infrastructure. The following is a list and description of the credentials Oracle Directory Integration Platform stores in this Credential Store Framework:

- The Oracle Directory Integration Platform user password. The password is created during installation, stored as read-only, and read by run-time operations.
- The JKS password. The JKS password is used if the Server Only (mode 2) SSL setting is configured for connecting to the Oracle back-end directory or a third-party directory. You can use the WebLogic Scripting Tool (WLST) `createCred()` command to write the keystore password to the Credential Store Framework. For example: after invoking the WLST shell and connecting to the Oracle WebLogic Admin Server using the `connect()` command, enter:

  ```
  createCred(map="dip", key="jksKey", type="PC", user="userName",
  password="password")
  ```

  The map and key options are fixed—the only supported values are `map="dip"` and `key="jksKey"`.

  You can use the wlst `listCred()` command to view the keystore password in the Credential Store Framework. For example: after invoking the WLST shell and connecting to the Oracle WebLogic Admin Server using the `connect()` command, enter:

  ```
  listCred(map="dip", key="jksKey")
  ```

See Also:
- The *Oracle Fusion Middleware Security Guide* for complete information about the Credential Store Framework of the Oracle Application Server 11g infrastructure.
- The *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference* for more information about the WLST commands.

Part II

General Administration of Oracle Directory Integration Platform

This part describes some of the general administrative tasks involved in running Oracle Directory Integration Platform. You can find more specific administrative information in the respective sections of this guide.

Part II contains the following chapters:

3

Administering Oracle Directory Integration Platform

This chapter describes tools you can use to administer Oracle Directory Integration Platform. It contains these topics:

- Graphical Tools for Administering Oracle Directory Integration Platform
- Command-Line Tools for Administering Oracle Directory Integration Platform

3.1 Graphical Tools for Administering Oracle Directory Integration Platform

You can use the following graphical tools to administer Oracle Directory Integration Platform:

- Using Fusion Middleware Control
- Using Oracle Internet Directory Self-Service Console

> **Note:** Prior to 11*g* Release 1 (11.1.1), the Oracle Directory Integration Platform was graphically administered by using the Oracle Directory Integration Server Administration tool. This tool is no longer available with the Oracle Directory Integration Platform. To graphically administer the Oracle Directory Integration Platform in 11*g* Release 1 (11.1.1) you must use Oracle Enterprise Manager Fusion Middleware Control.

3.1.1 Using Fusion Middleware Control

As of 11*g* Release 1 (11.1.1), you can graphically administer many Oracle Directory Integration Platform features from the Oracle Enterprise Manager Fusion Middleware Control. This console enables you to configure and manage all Oracle products from one user interface.

To use Oracle Enterprise Manager Fusion Middleware Control to administer Oracle Directory Integration Platform:

1. Connect to Oracle Enterprise Manager Fusion Middleware Control. The URL is of the form:

   `https://host:port/em`

2. In the left panel topology tree, expand the farm, then Identity and Access. Alternatively, from the farm home page, expand Fusion Middleware, then Identity

and Access. Oracle Directory Integration Platform components are listed in both places.

To distinguish one component from another, move the mouse over the component name and view the full name of the component in the tool tip.

3. Select the Oracle Directory Integration Platform component you want to manage.
4. Use the DIP Server menu to select tasks.

You can use the DIP Server menu to navigate to other Fusion Middleware Control pages for Oracle Directory Integration Platform.

3.1.1.1 The Oracle Directory Integration Platform Home Page

The Home Page for Oracle Directory Integration Platform in Oracle Enterprise Manager Fusion Middleware Control provides statistics and information about the component, including:

- The status of Oracle Directory Integration Platform components, such as the Quartz Scheduler and MBeans.
- The amount of CPU and memory being utilized.
- Information about existing Synchronization Profiles, including name, status, average execution time, and successful and failed propagation of changes.
- Information about existing Provisioning Profiles, including name, status, average execution time, and successful and failed propagation of changes.

3.1.2 Using Oracle Internet Directory Self-Service Console

The Oracle Internet Directory Self-Service Console enables you to delegate administrative privileges to various administrators and to users. It is a ready-to-use standalone application created with Oracle Delegated Administration Services that provides a single graphical interface for delegated administrators and users to manage data in the directory. The Oracle Internet Directory Self-Service Console enables both administrators and users, depending on their privileges, to perform various directory operations. In an integrated deployment, the Oracle Internet Directory Self-Service Console is primarily used for customizing realm parameters.

Note: Oracle Directory Integration Platform 11*g* Release 1 (11.1.1) interoperates with and supports Oracle Delegated Administration Services release 10.1.4.3.0.

See Also: *Oracle Fusion Middleware Guide to Delegated Administration for Oracle Identity Management*

3.2 Command-Line Tools for Administering Oracle Directory Integration Platform

The following command-line tools, located in the *ORACLE_HOME*/bin directory, are available for administering Oracle Directory Integration Platform:

Notes:

- Best security practice is to provide a password only in response to a prompt from the command.
- You must set the WLS_HOME and ORACLE_HOME environment variables before executing any of the Oracle Directory Integration Platform commands.
- Refer to the command-specific sections throughout this document and the *Oracle Identity Management User Reference* for additional information on each of the tools described in the following list.

- dipStatus: Allows you to check the status of Oracle Directory Integration Platform and whether or not it is registered. Refer to "Viewing the Status of Oracle Directory Integration Platform Using the dipStatus Utility" on page 4-4 for more information.
- manageDIPServerConfig: Manages Oracle Directory Integration Platform configuration settings including refresh interval, the Oracle back-end directory port number, keystore location and password, and the number of scheduler threads. Refer to "Managing Oracle Directory Integration Platform Using manageDIPServerConfig" on page 4-9 for more information.
- manageSyncProfiles: Manages Oracle Directory Integration Platform synchronization profiles. Refer to "Managing Synchronization Profiles Using manageSyncProfiles" on page 7-16 for more information.
- syncProfileBootstrap: Performs the initial migration of data between a connected target directory and the Oracle back-end directory based on a synchronization profile or LDIF file. Refer to "Directory Bootstrapping Using syncProfileBootstrap" on page 8-1 for more information.
- expressSyncSetup: Creates profiles for standard LDAP directories using prepackaged templates based on the directory type. Refer to "Creating Import and Export Synchronization Profiles Using expressSyncSetup" on page 17-2 for more information.
- provProfileBulkProv: Performs initial migration of data from an LDIF file to the Oracle back-end directory for a provisioning profile. Refer to "Bulk Provisioning Using the provProfileBulkProv Tool" on page 12-7 for more information.
- oidprovtool: Administers provisioning profile entries in the directory by enabling you to perform tasks such as:
  - Create new provisioning profiles
  - Enable or disable existing provisioning profiles
  - Modify existing provisioning profiles
  - Delete existing provisioning profiles
  - Get the current status of a provisioning profile

  - Clear all errors in an existing provisioning profile

  Refer to "Managing Provisioning Profiles Using oidprovtool" on page 13-2 for more information.

- `schemasync`: Directory Integration Platform does not support the synchronization of schema and ACLs. You can use the `schemasync` tool to identify differences in schema, specifically attributes and object classes, between the Oracle back-end directory and the connected directories. After identifying the differences, you can make the appropriate changes to the LDIF file containing the schema and then use the `ldapadd` and `ldapmodify` tools to upload the schema differences. The `schemasync` tool is located in the *ORACLE_HOME*/bin directory.

  See: *Oracle Fusion Middleware Reference for Oracle Identity Management* for more information.

3.2.1 Using Standard LDAP Command-Line Tools

Oracle Directory Integration Platform supports the standard LDAP command-line utilities, including those listed in Table 3–1.

For security reasons, avoid supplying a password on the command-line whenever possible. A password typed on the command line is visible on your screen and might appear in log files or in the output from the `ps` command. When you supply a password at a prompt, it is not visible on the screen, in `ps` output, or in log files. Use the `-q` and `-Q` options, respectively, instead of the `-P` *`password`* and `-w` *`password`* options.

The `LDAP` tools have been modified to disable the options `-w` *`password`* and `-P` *`password`* when the environment variable `LDAP_PASSWORD_PROMPTONLY` is set to `TRUE` or `1`. Use this feature whenever possible.

See Also: "Using Passwords with Command-Line Tools" in *Oracle Fusion Middleware Reference for Oracle Identity Management*.

Table 3–1 Entry and Attribute Management Command-Line Tools

Tool	Description
`catalog`	Indexes attributes. This tool is only supported if Oracle Internet Directory is your Oracle back-end directory.
`ldapadd`	Adds entries and their object classes, attributes, and values to the directory.
`ldapaddmt`	Supports multiple threads for concurrently adding entries and their object classes, attributes, and values to the directory. This tool is only supported if Oracle Internet Directory is your Oracle back-end directory.
`ldapbind`	Determines whether you can authenticate a client to a server.
`ldapcompare`	Matches specified attribute values with an entry's attribute values.
`ldapdelete`	Removes entries from the directory.
`ldapmoddn`	Modifies an entry's DN or RDN.
`ldapmodify`	Modifies an entry's attributes.

Table 3–1 (Cont.) Entry and Attribute Management Command-Line Tools

Tool	Description
`ldapmodifymt`	Supports multiple threads to modify entries concurrently. This tool is only supported if Oracle Internet Directory is your Oracle back-end directory.
`ldapsearch`	Searches for entries in the directory

See Also: *Oracle Identity Management User Reference* for the required syntax for each of the tools listed in Table 3–1.

4

Managing the Oracle Directory Integration Platform

This chapter discusses the Oracle Directory Integration Platform and explains how to configure and manage it. It contains these topics:

- Operational Information About the Oracle Directory Integration Platform
- Viewing Oracle Directory Integration Platform Status and Registration Information
- Managing Oracle Directory Integration Platform Using Fusion Middleware Control
- Starting and Stopping Oracle Directory Integration Platform Using WLST
- Managing Oracle Directory Integration Platform Using manageDIPServerConfig
- Configuring Oracle Directory Integration Platform for SSL Mode 2 Server-Only Authentication
- Managing the SSL Certificates of Back-End Directories and Connected Directories
- Oracle Directory Integration Platform in a High Availability Scenario
- Managing Oracle Directory Integration Platform in a Replicated Environment

See Also: "Oracle Directory Integration Platform" on page 1-5 for a summary of the functions performed by the Oracle Directory Integration Platform

Note: For security reasons, Oracle recommends that you run the Oracle Directory Integration Platform on the same host as the Oracle back-end directory. If you run Oracle Directory Integration Platform and the Oracle back-end directory on different hosts, Oracle recommends running them using SSL.

4.1 Operational Information About the Oracle Directory Integration Platform

This section introduces structural and operational information about the Oracle Directory Integration Platform and contains these topics:

- Directory Integration Profiles

4.1.1 Directory Integration Profiles

In Oracle Directory Integration Platform, you can create two types of profiles: a directory synchronization profile and a directory provisioning profile. A **directory synchronization profile** describes how synchronization is carried out between the Oracle back-end directory and a connected directory. You can create two types of directory synchronization profiles: an import profile and an export profile. An *import profile* imports changes from a connected directory to the Oracle back-end directory while an *export profile* exports changes from the Oracle back-end directory to a connected directory. A **directory provisioning profile** describes the nature of provisioning-related notifications that Oracle Directory Integration Platform sends to the directory-enabled applications. Sometimes a provisioning profile is also configured to notify the Oracle back-end directory about the changes happening in the application's data source. Multiple profiles can be used at the same time.

Each type of profile is special kind of **directory integration profile**, which is an entry in the Oracle back-end directory that describes how Oracle Directory Integration Platform communicates with external systems and what is communicated.

4.1.2 Oracle Directory Integration Platform Event Propagation in a Multimaster Oracle Back-end Directory Replication Environment

In a multimaster Oracle back-end directory environment, changes to directory synchronization profiles on one Oracle back-end directory node must be replicated or copied to any secondary nodes. This allows a directory synchronization profile to execute on a secondary node in the event of a problem on the primary node.

In a multimaster Oracle Universal Directory or Oracle Directory Server Enterprise Edition environment, if a suffix containing DIP meta-data is chosen for replication, the profiles are automatically replicated.

In a multimaster Oracle Internet Directory replication environment, however, changes to directory synchronization profiles on one Oracle Internet Directory node are not automatically replicated on other Oracle Internet Directory nodes. For this reason, you must copy the profiles on the primary node to any secondary nodes. For instructions, see the following section.

Note: The value assigned to the `orcllastappliedchangenumber` attribute in a directory synchronization profile is local to the Oracle Internet Directory node where the profile is located. This means that if you copy a directory synchronization profile from one Oracle Internet Directory node to another, the correct state of synchronization or event propagation will not be preserved.

4.1.2.1 Directory Synchronization in an Oracle Back-end Directory Multimaster Replication Environment

If you copy the profiles on the primary node to any secondary nodes, update the `lastchangenumber` attribute with the value from the target node, as follows. This step needs to be done once after the profile is set up.

This update is required if your Oracle back-end directory is Oracle Internet Directory. If your Oracle back-end directory is either Oracle Unified Directory or Oracle Directory Server Enterprise Edition, this step is only required if you copy the suffix containing DIP metadata from a primary node to secondary nodes instead of using replication.

1. Disable the synchronization profile.
2. Get the value of the `lastchangenumber` attribute on the target node using the `ldapsearch` command.
3. Use `ldapsearch` to get the LDIF dump of the profile entry.
4. Use `ldapadd` to add the profile to the other Oracle back-end directory instance.
5. Use the `updatechgnum` operation of the manageSyncProfiles command to update the `lastchangenumber` attribute in the export profile you copied to the target node with the value you obtained in Step 2.
6. Enable the synchronization profile.

4.1.2.2 Directory Provisioning in an Oracle Internet Directory Multimaster Replication Environment

In a default multimaster Oracle Internet Directory replication environment, the Oracle Directory Integration Platform is installed in the same location as the primary Oracle Internet Directory. If the primary node fails, event propagation stops for all profiles located on the node. Although the events are queued and not lost while the primary node is stopped, the events will not be propagated to any applications that expect them. To ensure that events continue to be propagated even when the primary node is down, you must copy the version 1.0 and version 2.0 directory provisioning profiles to other secondary nodes in a multimaster Oracle Internet Directory environment. Version 3.0 directory provisioning profiles are automatically replicated.

Note: Directory provisioning profiles should be copied from the primary node to any secondary nodes *only* immediately after an application is installed and before any user changes are made in Oracle Internet Directory.

To copy the directory provisioning profiles from a primary node to any secondary nodes, use the `update` operation of the manageSyncProfiles command.

See Also: The Oracle Directory Integration Platform chapter of *Oracle Identity Management User Reference* for more information on the manageSyncProfiles command.

4.2 Viewing Oracle Directory Integration Platform Status and Registration Information

This topic explains how to view Oracle Directory Integration Platform status and registration information and contains the following sections:

- Viewing the Status of Oracle Directory Integration Platform Using the dipStatus Utility
- Viewing Oracle Directory Integration Platform Registration Information Using the ldapsearch Utility

4.2.1 Viewing the Status of Oracle Directory Integration Platform Using the dipStatus Utility

The `dipStatus` utility, located in the *ORACLE_HOME*/bin directory, allows you to check the status of Oracle Directory Integration Platform and whether or not it is registered.

Notes:

- Best security practice is to provide a password only in response to a prompt from the command.
- You must set the `WLS_HOME` and `ORACLE_HOME` environment variables before executing any of the Oracle Directory Integration Platform commands.
- The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. Refer to the *Configuring SSL* chapter in *Oracle Fusion Middleware Securing Oracle WebLogic Server* for more information.

4.2.1.1 Syntax for dipStatus

dipStatus

```
dipStatus -h HOST -p PORT -D wlsuser [-ssl -keystorePath PATH_TO_KEYSTORE
-keystoreType TYPE] [-help]
```

4.2.1.2 Arguments for dipStatus

-h | -host

Oracle WebLogic Server where Oracle Directory Integration Platform is deployed.

-p | -port

Listening port of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed.

-D | -wlsuser

Oracle WebLogic Server login ID.

Note: You will be prompted for the Oracle WebLogic Server login password. You cannot provide the password as a command-line argument.

Best security practice is to provide a password only in response to a prompt from the command. If you must execute `dipStatus` from a script, you can redirect input from a file containing the Oracle WebLogic Server password. Use file permissions to protect the file and delete it when it is no longer necessary.

-ssl

Executes the command in SSL mode.

Note: The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. Refer to the *Configuring SSL* chapter in *Oracle Fusion Middleware Securing Oracle WebLogic Server* for more information.

-keystorePath

The full path to the keystore.

-keystoreType

The type of the keystore identified by `-keystorePath`. For example: `-keystorePath jks` or `-keystorePath PKCS12`

-help

Provides usage help for the command.

4.2.1.3 Examples for dipStatus

```
dipStatus -h myhost.mycompany.com -p 7005 -D login_ID

dipStatus -help
```

4.2.2 Viewing Oracle Directory Integration Platform Registration Information Using the ldapsearch Utility

To view registration information for the Oracle Directory Integration Platform component using the `ldapsearch` utility, perform a base search on its entry. For example:

```
ldapsearch -h oid_host -p port -D cn=orcladmin -q -s base -b
"cn=odisrv,cn=Registered Instances,cn=Directory Integration
Platform,cn=Products,cn=OracleContext" objectclass=*
```

Note: You will be prompted for the password.

This example search returns the following:

```
Dn: cn=odisrv,cn=Registered Instances,cn=Directory Integration
Platform,cn=Products,cn=OracleContext
userpassword: {SHA}+vk5wSvnVoXCBCRyBWJnH0S33zc=
orclaci: access to entry by self (add,delete,browse,proxy); access to attr=(*) by
self (search,read,write,compare)
orclversion: 3.0
cn: odisrv
objectclass: orclodiserver; top;
authpassword;oid: {SASL/MD5}2NOnGTWkSP9c1w7R/o9Djw==
{SASL/MD5-DN}ezUTC3k7rSL41ZxdxhlXxw==;{SASL/MD5-U}kEQcl+/AZEXVukeA5YPnog==
```

4.3 Managing Oracle Directory Integration Platform Using Fusion Middleware Control

This section describes how to use Oracle Enterprise Manager Fusion Middleware Control to manage Oracle Directory Integration Platform. It contains these topics:

- Viewing Oracle Directory Integration Platform Runtime Information Using Fusion Middleware Control
- Starting Oracle Directory Integration Platform with Fusion Middleware Control
- Stopping Oracle Directory Integration Platform with Fusion Middleware Control
- Managing the Oracle Directory Integration Platform Server Configuration
- Managing Oracle Directory Integration Platform Logging Using Fusion Middleware Control
- Auditing Oracle Directory Integration Platform Using Fusion Middleware Control

4.3.1 Viewing Oracle Directory Integration Platform Runtime Information Using Fusion Middleware Control

To view runtime information for the Oracle Directory Integration Platform component using Oracle Enterprise Manager Fusion Middleware Control:

1. Open a Web browser and enter the Oracle Enterprise Manager Fusion Middleware Control URL for your environment. The format of the Oracle Enterprise Manager Fusion Middleware Control URL is: https://*host*:*port*/em.
2. Log in to Oracle Enterprise Manager Fusion Middleware Control.
3. In the navigation panel on the left, click or expand the **Identity and Access** entry and then select the **DIP** component that you want to view runtime information for. Oracle Enterprise Manager Fusion Middleware Control opens the Oracle Directory Integration Platform home page, which includes the following information:
   - Synchronization Profiles: Summary of the configured synchronization profiles.
   - Provisioning Profiles: Summary of the configured provisioning profiles.
   - Resource Usage: Charts showing percentages of CPU and Memory being utilized on the Oracle Directory Integration Platform host.

     Tip: To return to the Oracle Directory Integration Platform home page after navigating to other Oracle Directory Integration Platform pages in Oracle Enterprise Manager Fusion Middleware Control, click **Home** on the **DIP Server** menu.

4.3.2 Starting Oracle Directory Integration Platform with Fusion Middleware Control

To start Oracle Directory Integration Platform by using Oracle Enterprise Manager Fusion Middleware Control:

1. Open a Web browser and enter the Oracle Enterprise Manager Fusion Middleware Control URL for your environment. The format of the Oracle Enterprise Manager Fusion Middleware Control URL is: https://*host*:*port*/em.
2. Log in to Oracle Enterprise Manager Fusion Middleware Control.
3. In the navigation panel on the left, click or expand the **Identity and Access** entry and then select the **DIP** component that you want to start.

4. Click the **DIP Server** menu, point to **Control**, and then click **Start**.

4.3.3 Stopping Oracle Directory Integration Platform with Fusion Middleware Control

To stop Oracle Directory Integration Platform by using Oracle Enterprise Manager Fusion Middleware Control:

1. Open a Web browser and enter the Oracle Enterprise Manager Fusion Middleware Control URL for your environment. The format of the Oracle Enterprise Manager Fusion Middleware Control URL is: https://*host*:*port*/em.
2. Log in to Oracle Enterprise Manager Fusion Middleware Control.
3. In the navigation panel on the left, click or expand the **Identity and Access** entry and then select the **DIP** component that you want to stop.
4. Click the **DIP Server** menu, point to **Control**, and then click **Stop**.
5. When the confirmation dialog appears, click **Yes**.

4.3.4 Managing the Oracle Directory Integration Platform Server Configuration

To configure the Oracle Directory Integration Platform Server Refresh Interval and settings for the connection to the Oracle back-end directory using Oracle Enterprise Manager Fusion Middleware Control:

1. Open a Web browser and enter the Oracle Enterprise Manager Fusion Middleware Control URL for your environment. The format of the Oracle Enterprise Manager Fusion Middleware Control URL is: https://*host*:*port*/em.
2. Log in to Oracle Enterprise Manager Fusion Middleware Control. Oracle Enterprise Manager Fusion Middleware Controls opens the Home Page.
3. In the navigation panel on the left, click or expand the **Identity and Access** entry and then select the **DIP** component that you want to manage.
4. Click the **DIP Server** menu, point to **Administration**, and then click **Server Properties**.

   The DIP Server Configuration page appears.

   The following list describes the fields and options on the DIP Server Configuration page:

   - **Server Refresh Interval (sec)**: The time interval (amount of time in seconds) that controls how often the Oracle Directory Integration Platform server refreshes profile configuration details.
   - **OID Connection Settings / OUD Connection Settings / ODSEE Connection Settings:** Enter the host name and port of the Oracle back-end directory where you want to save the Oracle Directory Integration Platform configuration.
   - **OID connect SSL Mode / OUD connect SSL Mode / ODSEE connect SSL Mode**: Specify the mode Directory Integration Platform uses to connect to the Oracle back-end directory.

Note: For Oracle Internet Directory, you cannot specify no-SSL (mode 0) as the mode Directory Integration Platform uses to connect to the Oracle back-end directory using Oracle Enterprise Manager Fusion Middleware Control.

For Oracle Unified Directory and Oracle Directory Server Enterprise Edition, you can specify no-SSL (mode 0).

The supported options are:

- No-auth (mode 1): Directory Integration Platform connects to the Oracle back-end directory using only SSL encryption.

  This option is only available if Oracle Internet Directory is your Oracle back-end directory. It is *not available* if Oracle Unified Directory or Oracle Directory Server Enterprise Edition is your Oracle back-end directory.

- Server Only (mode 2): Directory Integration Platform connects to and is authenticated only by the Oracle back-end directory.

Note: If you select the Server Only (mode 2) option, you must configure Oracle Directory Integration Platform for SSL Mode 2 server-only authentication from the command line. Refer to "Configuring Oracle Directory Integration Platform for SSL Mode 2 Server-Only Authentication" on page 4-11 for more information.

5. Optionally, click **Test Connection** to test the connection to the target Oracle back-end directory.
6. Make the desired changes and click the **Apply** button.

4.3.5 Managing Oracle Directory Integration Platform Logging Using Fusion Middleware Control

Oracle Enterprise Manager Fusion Middleware Control allows you to list, search, and configure log files across Oracle Fusion Middleware components. You can view log files from Oracle Enterprise Manager Fusion Middleware Control or download log files and view them using another tool. You can also list and search log files using the WLST command-line tool.

See Also: The *Oracle Fusion Middleware Administrator's Guide* for complete information on logging using Oracle Enterprise Manager Fusion Middleware Control.

4.3.6 Auditing Oracle Directory Integration Platform Using Fusion Middleware Control

Oracle Directory Integration Platform utilizes the Common Audit Framework of the Oracle Application Server 11g infrastructure for compliance, monitoring, and analytics purposes. Using Oracle Enterprise Manager Fusion Middleware Control, you can view, search, and manage audit data and event settings for Oracle Directory Integration Platform. Refer to the *Oracle Fusion Middleware Application Security Guide* for complete information on auditing.

4.4 Starting and Stopping Oracle Directory Integration Platform Using WLST

You can start and stop Oracle Directory Integration Platform from the command line using the WebLogic Scripting Tool (WLST) by connecting to the WebLogic Admin Server and executing the `startApplication("DIP")` and `stopApplication("DIP")` commands.

See:

- The *Oracle Fusion Middleware Oracle WebLogic Scripting Tool* for information on how to use the WLST command line tool.
- The *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference* for information WLST command tool syntax.

4.5 Managing Oracle Directory Integration Platform Using manageDIPServerConfig

The Manage DIP Server Configuration utility, `manageDIPServerConfig`, allows you to manage the Oracle Directory Integration Platform server configuration. The `manageDIPServerConfig` utility is located in the *ORACLE_HOME*`/bin` directory.

Notes:

- Best security practice is to provide a password only in response to a prompt from the command.
- You must set the `WLS_HOME` and `ORACLE_HOME` environment variables before executing any of the Oracle Directory Integration Platform commands
- The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. Refer to the *Configuring SSL* chapter in *Oracle Fusion Middleware Securing Oracle WebLogic Server* for more information.

4.5.1 Syntax for manageDIPServerConfig

manageDIPServerConfig

```
manageDIPServerConfig {get | set} -h HOST -p PORT -D wlsuser -attribute {sslmode |
refreshinterval | quartzthreadcount | quartzdbretryinterval | oidhostport |
keystorelocation} [-ssl -keystorePath PATH_TO_KEYSTORE -keystoreType TYPE]
[-value ATTRIBUTE_VALUE] [-help]
```

4.5.2 Arguments for manageDIPServerConfig

get | set

Operation to perform.

- get: Displays the current value of the config parameter in DIP configuration file
- set: Updates the value of the config parameter in DIP configuration file.

-h | -host

Oracle WebLogic Server host where Oracle Directory Integration Platform is deployed

-p | -port

Listen port of Oracle WebLogic Managed Server where Oracle Directory Integration Platform application is deployed.

-D | -wlsuser

Oracle WebLogic Server login ID.

> **Note:** You will be prompted for the Oracle WebLogic Server login password. You cannot provide the password as a command-line argument. A best security practice is to provide a password only in response to a prompt from the command. If you must execute `manageDIPServerConfig` from a script, you can redirect input from a file containing the Oracle WebLogic Server login password. Use file permissions to protect the file and delete it when it is no longer necessary.

-attr | -attribute

Identifies the attribute that `manageDIPServerConfig` performs the operation on. The following is a list and description of the attributes `manageDIPServerConfig` can perform operations on:

- `sslmode`: The SSL mode Oracle Directory Integration Platform uses to connect to the Oracle back-end directory. Supported values are 1 and 2. Use 1 to connect to the Oracle back-end directory using SSL Mode 1 (No Authentication). (SSL Mode 1 is only supported if Oracle Internet Directory is your Oracle back-end directory.) Use 2 to connect to the Oracle back-end directory using SSL Mode 2 (Server Only Authentication).
- `refreshinterval`: The time interval (amount of time in seconds) that controls how often the Oracle Directory Integration Platform server refreshes profile configuration details.
- `quartzthreadcount`: Controls how many profiles can be scheduled in parallel. The default value is 15. If you have more than 15 profiles, increase the `quartzthreadcount` attribute accordingly.
- `quartzdbretryinterval`: Controls how often Oracle Directory Integration Platform's Quartz scheduler attempts to reconnect to the Oracle back-end directory database.
- `oidhostport`: Identifies the host and port of the Oracle back-end directory associated with Oracle Directory Integration Platform. Specify values for the `oidhostport` attribute in the form of `host:port`.
- `keystorelocation`: Specifies the absolute path to the Java Keystore (JKS) based on the host where Oracle Directory Integration Platform is deployed. When you specify the value for the `keystorelocation` attribute, be sure you use the appropriate path separators (that is, `/` for UNIX and Linux platforms, and `\` for Windows platforms).

-ssl

Executes the command in SSL mode.

> **Note:** The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. Refer to the *Configuring SSL* chapter in *Oracle Fusion Middleware Securing Oracle WebLogic Server* for more information.

-keystorePath

The full path to the keystore.

-keystoreType

The type of the keystore identified by `-keystorePath`. For example: `-keystorePath jks` or `-keystorePath PKCS12`

-val | -value

The value to set for the attribute This parameter is required with the set operation.

-help

Provides usage help for the command.

4.5.3 Tasks and Examples for manageDIPServerConfig

```
manageDIPServerConfig get -h myhost.mycompany.com -p 7005 -D login_ID \
   -attr sslmode

manageDIPServerConfig set -h myhost.mycompany.com -p 7005 -D login_ID \
   -attr sslmode -val 2

manageDIPServerConfig set -h myhost.mycompany.com -p 7005 -D login_ID \
   -attr oidhostport -value OID_host:OID_SSL_port
```

4.6 Configuring Oracle Directory Integration Platform for SSL Mode 2 Server-Only Authentication

For instructions about how to configure DIP for SSL authentication with directories *other than* Oracle Internet Directory, see Section 4.6.3. Otherwise, before configuring Oracle Directory Integration Platform to use SSL mode in Section 4.6.2, ensure that the Oracle back-end directory is configured for SSL Server-Auth authentication in Section 4.6.1.

> **Note:** The following information describes SSL configuration for a single component. If you are configuring SSL for multiple components, you can use the Oracle SSL Automation Tool, which enables you to configure SSL for multiple components using a domain-specific CA.
>
> Refer to the *Oracle Fusion Middleware Administrator's Guide* for complete information about the Oracle SSL Automation Tool.

4.6.1 To Configure Oracle Internet Directory for SSL Server-Auth Authentication

Complete the following steps before configuring the Oracle Directory Integration Platform software to use SSL mode. If you have already configured the Oracle Internet Directory software for SSL authentication, skip this section and proceed to Section 4.6.2.

Oracle recommends creating a new OID component and configuring it for SSL server-authentication mode instead of changing the default configuration of oid1.

1. Create a new Oracle Internet Directory component.

   Follow the steps in the "Creating an Oracle Internet Directory Component by Using opmnctl" section, which is located in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

   Name the new Oracle Internet Directory component oid2 (or something similar).

2. Configure SSL for the new Oracle Internet Directory component (oid2).

   Follow the steps in the "Configuring SSL by Using Fusion Middleware Control" section, which is located in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

4.6.2 To Configure the Oracle Directory Integration Platform for SSL Authentication

This procedure describes how to configure the Oracle Directory Integration Platform for SSL authentication with Oracle Internet Directory. For instructions about how to configure Oracle Directory Integration Platform for SSL authentication with other directories, see Section 4.6.3, "To Configure Oracle Directory Integration Platform for SSL Authentication With Directories Other Than OID," on page 4-14.

Before you begin, verify that Oracle Internet Directory is configured for SSL Server-Auth authentication. If necessary, complete the steps in Section 4.6.1 before attempting the steps in this section.

1. Run the following command to export the trusted certificate from the Oracle Internet Directory wallet.

   `orapki wallet export -wallet` *Path_to_OID_wallet* `-dn` *Subject_DN_of_trusted_certificate* `-cert` *path_to_certificate_file*

   The Oracle Internet Directory wallet is available in the following location when created using the Fusion Middleware user interface: `$ORACLE_INSTANCE/OID/admin/`*wallet_name*

   For example:

   ```
   orapki wallet export
   -wallet /home/Middleware/asinst_1/OID/admin/oidwallet
   -dn "cn=ldap.oracle.com"
   -cert /home/Middleware/asinst_1/OID/admin/oidcert.txt
   ```

2. Create a Java Keystore (JKS) using the keytool, and import the trusted certificate exported in the previous step into the JKS.

   `keytool -importcert -trustcacerts -alias` *Some_alias_name* `-file` *Path_to_certificate_file* `-keystore` *path_to_keystore*

   For example:

   ```
   keytool -importcert -trustcacerts -alias OID2
   -file /home/Middleware/asinst_1/OID/admin/oidcert.txt
   -keystore /home/Middleware/dip.jks
   ```

The system will prompt for a keystore password. Type a new password for this keystore.

Notes:

- If you use the `-keystore` option and the keystore does not exist, keytool creates the keystore.

3. Run the following command to update the Java Keystore location in Oracle Directory Integration Platform.

   `manageDIPServerConfig set -attr keystorelocation -val` *full_path_to_keystore* `-h` *weblogic_host* `-p` *weblogic_managed_server_port* `-wlsuser` *weblogic_user*

   Note: *full_path_to_keystore* represents the absolute path to the Java Keystore (JKS) based on the host where Oracle Directory Integration Platform is deployed. When you specify the absolute path to the JKS, use the appropriate path separators (that is, / for UNIX and Linux platforms, and \ for Windows platforms).

   For example:

   ```
   manageDIPServerConfig set -attr keystorelocation
   -val /home/Middleware/dip.jks -h host -p 7005
   -wlsuser weblogic
   ```

   The system will prompt for the WebLogic password.

4. Run the following commands to create a CSF credential and update the Java Keystore password:

   a. Open the WLST prompt by running the following command:

      ```
      $ORACLE_HOME/common/bin/wlst.sh
      ```

   b. Connect to the WebLogic Admin Server:

      `connect('`*Weblogic_User*`', '`*Weblogic_password*`', 't3://`*Weblogic_Host*`:`*Weblogic_AdminServer_Port*`')`

   c. Create the credential and update the Java Keystore password:

      `createCred(map="dip", key="jksKey", user="jksuser", password="`*JKS_password_created_previously_in_step_2*`")`

5. Log in to the Fusion Middleware user interface and update the Oracle Directory Integration Platform SSL configuration.

   Choose **DIP > Server Properties**, then set SSL Mode to 2 and the port value to the Oracle Internet Directory SSL port.

6. Restart the Oracle WebLogic managed server.

   Oracle Directory Integration Platform will now connect to Oracle Internet Directory in SSL Server authentication mode.

4.6.3 To Configure Oracle Directory Integration Platform for SSL Authentication With Directories Other Than OID

This section describes how to configure Oracle Directory Integration Platform for SSL authentication with non-OID directories, including Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server), and Oracle Universal Directory (OUD).

1. Export the trusted certificate from the directory and save it to a file.
2. Import the trusted certificate from the directory into the Java Keystore (JKS).

   `keytool -importcert -trustcacerts -alias` *Some_alias_name* `-file` *Path_to_certificate_file* `-keystore` *path_to_keystore*

   For example:

   ```
   keytool -importcert -trustcacerts -alias sunone
   -file /home/Middleware/sunone.cert -keystore
   /home/Middleware/dip.jks
   ```

   Notes:

   - If you use the `-keystore` option and the keystore does not exist, keytool creates the keystore.

3. During profile creation, select the SSL option and provide the third-party directory SSL port.

4.7 Managing the SSL Certificates of Back-End Directories and Connected Directories

The Oracle Directory Integration Platform can use SSL to connect the Oracle back-end directory and connected directories. When using SSL with no authentication to connect to the Oracle back-end directory, no certificate is required. However, when connecting to the Oracle back-end directory using SSL with server authentication, you need a trust-point certificate to connect to the LDAP server. The Oracle Directory Integration Platform expects the certificate to be in a Java Keystore (JKS).

You can use the `manageDIPServerConfig` command with the `keystorelocation` argument to manage the keystore location and you can use the WLST Credential Store commands with `map="dip"` and `key="jksKey"` to manage the keystore password.

See Also:

- "Managing Oracle Directory Integration Platform Using manageDIPServerConfig" for more information about the `manageDIPServerConfig` command.
- *Oracle Fusion Middleware Administrator's Guide* for more information about managing keystores using WLST.

4.7.1 Detecting and Removing an Expired Certificate

You can use the keytool utility in the $*JAVA_HOME*/bin directory to detect and remove expired certificates for Oracle Directory Integration Platform.

To list the valid dates for a trusted certificate in the keystore, execute the keytool utility as follows:

```
$JAVA_HOME/bin/keytool -list -v -keystore PATH_TO_KEYSTORE
```

To delete a trusted certificate from the keystore, execute the keytool utility as follows:

```
$JAVA_HOME/bin/keytool -delete -alias mycert -keystore PATH_TO_KEYSTORE
```

Note: You will be prompted for the password to the keystore while executing these commands.

For general information about certificate expiration, see Chapter 7, "Managing Keystores, Wallets, and Certificates," of the *Oracle Fusion Middleware Administrator's Guide*.

4.8 Oracle Directory Integration Platform in a High Availability Scenario

In a high availability architecture, Oracle Directory Integration Platform is deployed on a Oracle WebLogic Cluster that has at least two servers as a part of the cluster. The Oracle WebLogic Server starts, stops and monitors Oracle Directory Integration Platform in the cluster. By default, Oracle Directory Integration Platform leverages the high availability features of the underlying Oracle WebLogic Clusters. In case of hardware or other failures, session state is available to other cluster nodes that can resume the work of the failed node.

In a high availability environment, Node Manager is configured to monitor the Oracle WebLogic Servers. In case of failure, Node Manager restarts the Oracle WebLogic Server. If Node Manager cannot restart the server, then the front-ending load balancing router detects failure of a WebLogic instance in the Cluster and routes traffic to surviving instances.

When the Oracle back-end directory is deployed in an active-active high availability configuration, all the Oracle back-end directory instances belonging to the cluster share the same database. Any changes made to Oracle Directory Integration Platform on one Oracle back-end directory node would automatically be propagated to all the Oracle back-end directory instances in the cluster.

See: *Oracle Fusion Middleware High Availability Guide* for complete information on Oracle Directory Integration Platform in a high availability scenario.

4.9 Managing Oracle Directory Integration Platform in a Replicated Environment

For provisioning and synchronization, the replicated directory is different from the master directory. Any profiles created in the original directory need to be re-created in the new directory, and all configurations must be performed as in the original directory.

Part III

Synchronization Using Oracle Directory Integration Platform

This part discusses the concepts and components involved in synchronization between the Oracle Directory Integration Platform and other identity management systems. It also discusses things you should consider when deciding how to deploy synchronization.

5

Understanding the Oracle Directory Synchronization Service

This chapter discusses the synchronization profiles and connectors that link the Oracle back-end directory and connected directories. It contains these topics:

- Components Involved in Oracle Directory Synchronization
- How Synchronization Works

> **See Also:** Chapter 1, "Introduction to Oracle Identity Management Integration" for a conceptual discussion of Oracle Directory Integration Platform

5.1 Components Involved in Oracle Directory Synchronization

This section discusses the components involved in Oracle Directory synchronization. It contains these topics:

- Connectors for Directory Synchronization
- Directory Synchronization Profiles

5.1.1 Connectors for Directory Synchronization

To synchronize between the Oracle back-end directory and a connected directory, Oracle Directory Integration Platform relies on a prepackaged connectivity solution called a connector. Minimally, this connector consists of a **directory integration profile** containing all the configuration information required for synchronization.

5.1.1.1 Using Connectors with Supported Interfaces

When synchronizing between the Oracle back-end directory and a connected directory, Oracle Directory Integration Platform uses one of these interfaces: DB, LDAP, tagged, or LDIF. If the connected directory uses one of these interfaces, then the connector requires only a directory integration profile for synchronization to occur. For example, the Oracle Directory Server Enterprise Edition (Sun Java System Directory Server) connector provided with Oracle Internet Directory uses the LDAP interface to read the changes from Oracle Directory Server Enterprise Edition. The changes are in the format specific to Oracle Directory Server Enterprise Edition and can be determined by running `ldapsearch` in Oracle Directory Server Enterprise Edition.

5.1.1.2 Using Connectors Without Supported Interfaces

If a connected directory cannot use one of the interfaces supported by Oracle Directory Integration Platform, then, in addition to the directory integration profile, it requires

an agent. The agent transforms the data from one of the formats supported by Oracle Directory Integration Platform into one supported by the connected directory. An example is the Oracle Human Resources connector. It has both a prepackaged integration profile and an Oracle Human Resources agent. To communicate with the Oracle back-end directory, the agent uses the tagged file format supported by Oracle Directory Integration Platform. To communicate with the Oracle Human Resources system, the agent uses SQL (through an OCI interface).

5.1.2 Directory Synchronization Profiles

A directory integration profile for synchronization, called a **directory synchronization profile**, contains all the configuration information required for synchronization including:

- Direction of synchronization

  Some connected directories only receive data *from* the Oracle back-end directory—that is, they participate in export operations only. Others only supply data *to* the Oracle back-end directory—that is, they participate in import operations only. Still others participate in both import and export operations.

  A separate profile is used for each direction—that is, one profile for information coming into the Oracle back-end directory, and another for information going from the Oracle back-end directory to the connected directories.

- Type of interface

  Some connected directories can receive data in any of the interfaces built into Oracle Directory Integration Platform. These interfaces include LDAP, tagged, DB (for read-only), and LDIF. For these connected directories, the Oracle Directory Synchronization Service performs the synchronization itself directly, using the information stored in the profile.

- Mapping rules and formats

  In a directory synchronization environment, a typical set of entries from one domain can be moved to another domain. Similarly, a set of attributes can be mapped to another set of attributes.

  Mapping rules govern the conversion of attributes between a connected directory and the Oracle back-end directory. Each connector stores a set of these rules in the `orclodipAttributeMappingRules` attribute of its synchronization profile. The Oracle Directory Integration Platform uses these rules to map attributes as needed when exporting from the directory and interpreting data imported from a connected directory or file. When the Oracle Directory Integration Platform imports changes into the Oracle back-end directory, it converts the connected directory's change record into an LDAP change record following the mapping rules. Similarly, during export, the connector translates the Oracle back-end directory changes to the format understood by the connected directory.

- Connection details of the connected directory

  These details include such information about the connected directory as host, port, mode of connection—that is, either SSL or non-SSL—and the connected directory credentials.

- Other information

  Although the synchronization profile stores most of the information needed by a connector to synchronize the Oracle back-end directory with connected

directories, some connectors may need more. This is because some operations require additional configuration information at runtime.

You can store additional connector configuration information wherever and however you want. However, Oracle Directory Integration Platform enables you to store it in the synchronization profile as an attribute called `orclODIPAgentConfigInfo`. Its use is optional—that is, if a connector does not require such information, then leave this attribute empty.

The configuration information can pertain to the connector, the connected directory, or both. The Oracle back-end directory and Oracle Directory Integration Platform do not modify this information. When the connector is invoked, the Oracle Directory Integration Platform provides it with the information in this attribute as a temporary file.

> **See Also:** The attribute reference chapter of the *Oracle Identity Management User Reference* for a list and descriptions of the attributes in a directory integration profile

5.2 How Synchronization Works

Depending on where the changes are made, synchronization can occur:

- From a connected directory to the Oracle back-end directory
- From the Oracle back-end directory to a connected directory
- In both directions

Regardless of the direction in which the data flows, it is assumed that:

- During synchronization, incremental changes made on one directory are propagated to the other
- Once synchronization is complete, the information is maintained in both directories in the same manner

This section contains these topics:

5.2.1 Synchronizing from the Back-end Directory to a Connected Directory

The Oracle back-end directory maintains a change log in which it stores incremental changes made to directory objects. It stores these changes sequentially based on the change log number.

Synchronization from the Oracle back-end directory to a connected directory makes use of this change log. Consequently, when running the Oracle Directory Integration Platform, you must start the Oracle back-end directory with the default setting in which change logging is enabled.

Each time the Oracle Directory Synchronization Service processes a synchronization profile, it:

1. Retrieves the latest change log number up to which all changes have been applied.
2. Checks each change log entry more recent than that number.

3. Selects changes to be synchronized with the connected directory by using the filtering rules in the profile.
4. Applies the mapping rules to the entry and makes the corresponding changes in the connected directory.

The appropriate entries or attributes are then updated in that connected directory. If the connected directory does not use DB, LDAP, tagged, or LDIF formats directly, then the agent identified in its profile is invoked. The number of the last change successfully used is then stored in the profile.

Periodically, the Oracle back-end directory purges the change log after all profiles have used what they need, and identifies where subsequent synchronization should begin.

Note: To log all information for a synchronization profile, including entries that are synchronized, set the log level for the profile to All using Oracle Enterprise Manager Fusion Middleware Control or set the odip.profile.debuglevel parameter to a value of 63 using the `manageSyncProfiles` command.

5.2.2 Synchronizing from a Connected Directory to the Back-end Directory

When a connected directory uses DB, LDAP, tagged, or LDIF formats directly, changes to its entries or attributes can be automatically synchronized by the Oracle Directory Synchronization Service. Otherwise, the connector has an agent in its synchronization profile, which writes the changes to a file in the LDIF or tagged format. The Oracle Directory Synchronization Service then uses this file of connected directory data to update the Oracle back-end directory.

5.2.3 Synchronizing Directories with Interfaces Not Supported by the Back-end Directory

Some connected directories cannot receive data by using any of the interfaces supported by the Oracle back-end directories. Profiles for this type of directory contain an attribute identifying an "agent," which is a separate program for synchronization. The agent translates between the connected directory's unique format and a DB, LDAP, tagged, or LDIF file containing the synchronization data. The agent, as identified in the profile, is invoked by the Oracle Directory Synchronization Service.

When exporting data from the Oracle back-end directory to this type of connected directory, the Oracle Directory Synchronization Service creates the necessary file in the tagged or LDIF format. The agent then reads that file, translates it into the correct format for the receiving connected directory, and stores the data in that directory.

When importing data from this type of connected directory to the Oracle back-end directory, the agent creates the necessary tagged or LDIF format file. The Oracle Directory Synchronization Service then uses this file data to update the Oracle back-end directory.

6

Configuring Directory Synchronization

This chapter explains how to configure directory synchronization and how to format mapping rules. It contains these topics:

- Registering Connectors in Oracle Directory Integration Platform
- Synchronization Profile Templates
- Configuring Connection Details
- Configuring Mapping Rules
- Extending Mappings Using Custom Plug-ins
- Configuring Matching Filters
- Location and Naming of Files

> **See Also:** Chapter 3, "Administering Oracle Directory Integration Platform" for information on using Oracle Enterprise Manager Fusion Middleware Control.

6.1 Registering Connectors in Oracle Directory Integration Platform

Before deploying a connector, register it in the Oracle back-end directory that you are using with Oracle Directory Integration Platform. This registration involves creating a synchronization profile, which is stored as an entry in the directory. Refer to "Creating Synchronization Profiles" on page 7-1 for information about creating a directory synchronization profile using Oracle Enterprise Manager Fusion Middleware Control.

> **See Also:** "Directory Synchronization Profiles" on page 5-2

Attributes in a synchronization profile entry belong to the object class `orclodiProfile`. The only exception is the `orclodiplastappliedchangenumber` attribute, which belongs to the `orclchangesubscriber` object class.

The `2.16.840.1.113894.7` object identifier prefix is assigned to platform-related classes and attributes.

If your Oracle back-end directory is Oracle Internet Directory, the various synchronization profile entries in the directory are created under the following container:

```
cn=subscriber profile,cn=changelog subscriber,cn=oracle internet directory
```

For example, a connector called `OracleHRAgent` is stored in the directory as follows:

```
orclodipagentname=OracleHRAgent,cn=subscriber profile,cn=changelog
subscriber,cn=oracle internet directory
```

If your Oracle back-end directory is Oracle Unified Directory or Oracle Directory Server Enterprise Edition, the various synchronization profile entries in the directory are created under the following container:

```
cn=subscriber profile,cn=changelog subscriber,cn=Directory Integration Platform,
<suffix>
```

The <suffix> is input that you provided during DIP configuration with Oracle Unified Directory or Oracle Directory Server Enterprise Edition.

6.2 Synchronization Profile Templates

When you install Oracle Directory Integration Platform, template profiles are created for synchronization with the different directory types, including:

- Microsoft Active Directory 2003
- Microsoft Active Directory Lightweight Directory Service (AD LDS) Version 1 (previously known as Active Directory Application Mode or ADAM)
- IBM Tivoli Directory Server 6.2
- Sun Java System Directory Server 6.3 (going forward, renamed to Oracle Directory Server Enterprise Edition)
- Oracle Directory Server Enterprise Edition 11.1.1.3 (previously known as Sun Java System Directory Server)
- Novell eDirectory 8.8
- OpenLDAP-2.2
- Oracle Database
- LDIF files
- Tagged files

The property and mapping files used to create the template profiles are available in the $*ORACLE_HOME*`/ldap/odi/conf` directory.

6.3 Configuring Connection Details

You can configure the connection details for a third-party directory by creating or editing a synchronization profile using Oracle Enterprise Manager Fusion Middleware Control. To use one of the sample synchronization profiles that was creating during installation, be sure to specify the correct connection details. In addition to specifying the connection details, you must also ensure that the user account in the third-party directory has the necessary privileges to read user and group information.

You can also create the profiles based on the template properties file provided during installation. If you are doing this, then you must specify the connection details in the `odip.profile.condirurl`, and `odip.profile.condiraccount` properties of the profile. You will be prompted for the password.

In addition to specifying the connection details, you must also ensure that the user account in the third-party directory has the necessary privileges to read user and group information.

Each third-party directory requires a different configuration for getting deleted entries. Refer to the third-party directory's documentation to set up the tombstone configuration and privileges required to read tombstone entries. For example, with Microsoft Active Directory, you must also ensure that the user account has the privileges to replicate directory changes for every domain of the forest monitored for changes. You can do this by one of the following methods:

- Grant to this account Domain Administrative permissions
- Make this account a member of the Domain Administrator's group
- Grant to this account Replicating Directory Changes permissions for every domain of the forest that is monitored for changes

To grant this permission to a non-administrative user, follow the instructions in the "More Information" section of the Microsoft Help and Support article "How to Grant the 'Replicating Directory Changes' Permission for the Microsoft Metadirectory Services ADAM Service Account" available at `http://support.microsoft.com/`.

Some of the most important pieces of a directory synchronization profile include the connection details you assign to the properties listed in Table 6–1:

Table 6–1 Connection Detail Properties

Property	Description
`odip.profile.condirurl`	The URL of the connected directory: ■ To connect to an LDAP directory, use the form *host*:*port* ■ To connect in SSL mode, use the form *host*:*port*:1. ■ To connect to a database, use the form *host*:*port*:*sid*
`odip.profile.condiraccount`	The DN or account name used to connect to the third-party directory

Notes:

- The account information you specify must have sufficient privileges in the directory to which you are connecting.
- The account name is not required if you are using the LDIF or tagged data formats.
- You will be prompted for a password.

6.4 Configuring Mapping Rules

This section discusses how to configure mapping rules. It contains these topics:

- Distinguished Name Mapping
- Attribute-Level Mapping
- Manually Creating New Mapping Files
- Supported Attribute Mapping Rules and Examples
- Example: Mapping File for a Tagged-File Interface
- Example: Mapping Files for an LDIF Interface
- Updating Mapping Rules

You use the mapping rules attribute to specify how to convert entries from the source to the destination. The Oracle back-end directory must either be the source or the destination. When converting the entries, there are three types of mapping rules: domain rules, attribute rules, and reconciliation rules. These mapping rules allow you to specify distinguished name mapping, attribute-level mapping, and reconciliation rules. Note that reconciliation rules are only used with Novell eDirectory and OpenLDAP. For more information on using reconciliation rules, see Chapter 22, "Integrating with Novell eDirectory or OpenLDAP".

> **Note:** For information about configuring mapping rules that connect to Oracle Database, see Section 9.2, "Preparing the Mapping File" in the "Synchronizing with Tables in Oracle Database" chapter.

Mapping rules are organized in a fixed, tabular format, and you must follow that format carefully. Each set of mapping rules appears between a line containing only the word *DomainRules* or *AttributeRules* and a line containing only three number signs (###).

```
DomainRules
srcDomainName1: [dstDomainName1]: [DomainMappingRule1]
srcDomainName2: [dstDomainName2]: [DomainMappingRule2]
[DomainExclusionList]
srcDomainForExclusion1
srcDomainforExclustion2
###
AttributeRules
srcAttrName1:[ReqAttrSeq]:[SrcAttrType]:[SrcObjectClass]:[dstAttrName1]:
[DstAttrType]:[DstObjectClass]:[AttrMappingRule1]
srcAttrName1,srcAttrName2:[ReqAttrSeq]:[SrcAttrType]:[SrcObjectClass]:
[dstAttrName2]:[DstAttrType]:[DstObjectClass]:[AttrMappingRule2]
[AttributeExclusionList]
exclusionAttribute1
exclusionAttribute2
###
```

The expansion of `srcAttrName1` and `srcAttrName2` in the preceding example should be on a single, unwrapped long line.

6.4.1 Distinguished Name Mapping

This section specifies how entries are mapped between the Oracle back-end directory and a connected directory. If the mapping is between the Oracle back-end directory and an LDAP directory, then you can create multiple mapping rules. The domain rule specifications appear after a line containing only the keyword `DomainRules`. Each domain rule is represented with the components, separated by colons, and are described in Table 6–2.

Table 6–2 Domain Rule Components

Component Name	Description
`SrcDomainName`	Name of the domain or container of interest. Specify NONLDAP for sources other than LDAP and LDIF.

Table 6–2 (Cont.) Domain Rule Components

Component Name	Description
DstDomainName	Name of the domain of interest in the destination. Specify this component if the container for the entries in the destination directory is different from that in the source directory. If the value assigned to SrcDomainName is an LDAP or LDIF domain, then this field assumes the same value. However, if the value assigned to SrcDomainName is not an LDAP or LDIF domain, you must specify the container where entries should be created. If not specified, this field assumes the value of SrcDomainName under valid conditions. For destinations other than LDAP and LDIF, specify NONLDAP. Because "import" and "export" always refer to the Oracle back-end directory, a combination of NONLDAP:NONLDAP is not allowed.
DomainMappingRule	This rule is used to construct the destination DN from the source domain name, from the attribute given in AttributeRules, or both. This field is typically in the form of cn=%,l=%,o=oracle,dc=com. These specifications are used to put entries under different domains or containers in the directory. In the case of non-LDAP sources, this rule specifies how to form the target DN so it can add entries to the directory. This field is meaningful only when importing to the Oracle back-end directory, or when exporting to an LDIF file or another external LDAP-compliant directory. Specify this component if any part of an entry's DN in the destination directory is different from that in the source directory entry. This component is optional for LDAP-to-LDIF, LDAP-to-LDAP, or LDIF-to-LDAP synchronizations. If it is not specified, then the source domain and destination domain names are considered to be the same.

Example 6–1 Example of Distinguished Name Mapping

```
Distinguished Name Rules
%USERBASE INSOURCE%:%USERBASE ATDEST%:
```

USERBASE refers to the container from which the third-party directory users and groups must be mapped. Usually, this is the users container under the root of the third-party directory domain.

Example 6–2 Example of One-to-One Distinguished Name Mapping

For one-to-one mapping to occur, the DN in the third-party directory must match that in the Oracle back-end directory. In this example, the DN in the third-party directory matches the DN in the Oracle back-end directory. More specifically:

- The third-party directory host is in the domain us.mycompany.com, and, accordingly, the root of the third-party directory domain is us.mycompany.com. A user container under the domain would have a DN value cn=users,dc=us,dc=mycompany,dc=com.
- The Oracle back-end directory has a default realm value of dc=us,dc=mycompany,dc=com. This default realm automatically contains a users container with a DN value cn=users,dc=us,dc=mycompany,dc=com.

Because the DN in the third-party directory matches the DN in the Oracle back-end directory, one-to-one distinguished name mapping between the directories can occur.

If you plan to synchronize only the `cn=users` container under `dc=us,dc=mycompany,dc=com`, then the domain mapping rule is:

```
Distinguished Name Rules
cn=users,dc=us,dc=mycompany,dc=com:cn=users,dc=us,dc=mycompany,dc=com
```

This rule synchronizes every entry under `cn=users,dc=us,dc=mycompany,dc=com`. However, the type of object synchronized under this container is determined by the attribute-level mapping rules that follow the DN Mapping rules.

If you plan to synchronize the entry `cn=groups,dc=us,dc=mycompany,dc=com` under `cn=users,dc=us,dc=mycompany,dc=com` then the domain mapping rule is as follows:

```
cn=groups,dc=us,dc=mycompany,dc=com:cn=groups,cn=users,dc=us,dc=mycompany,dc=com
```

6.4.1.1 Excluding Domains

You can insert the `DomainExclusionList` header in map files and identify domains to be excluded during bootstrap and synchronization. Domains listed in the `DomainExclusionList` will be excluded during bootstrap and synchronization.

> **Note:** The distinguished names (DNs) listed in the `domainexclusionlist` identify the DNs of the containers in the source directory.

The following is an example of the `DomainExclusionList` header with example domains to exclude:

```
DomainExclusionList
OU=myou,OU=test,DC=mycompany,DC=com
OU=mynewou,OU=test,DC=mycompany,DC=com
```

Example 6–3, "Example Map File Using DomainExclusionList and AttributeExclusionList Headers" shows an example map file that includes the `DomainExclusionList` header. In this example, the entries under `RDN OU=myou, OU=rfi6894748, DC=imtest, DC=com and OU=mynewou, OU=rfi6894748, DC=imtest, DC=com` will be excluded.

6.4.2 Attribute-Level Mapping

The attribute rule specifications appear after a line containing only the keyword `AttributeRules`. Attribute rules specify how property values for an entry are related between two LDAP directories. For example, the `cn` attribute of a user object in one directory can be mapped to the `givenname` object in another directory. Similarly, the `cn` attribute of a group object in one directory can be mapped to the `displayname` attribute in another directory. Each attribute rule is represented with the components, separated by colons, and are described in Table 6–3. The attribute rule specifications end with a line three number signs (`###`).

Table 6–3 Components in Attribute Rules

Component Name	Description
`SrcAttrName`	For LDAP-compliant directory repositories, this parameter refers to the name of the attribute to be translated. For Oracle Database repositories, it refers to the `ColumnName` in the table specified by the `SrcClassName`. For other repositories this parameter can be appropriately interpreted.
`ReqAttrSeq`	Indicator of whether the source attribute must be passed to the destination. When entries are synchronized between the Oracle back-end directory and the connected directory, some attributes need to be used as synchronization keys. This field indicates whether the specified attribute is being used as a key. If so, regardless of whether the attribute has changed or not, the value of the attribute is extracted from the source. A nonzero integer value should be placed in this field if the attribute needs to be always passed on to the other end.
`SrcAttrType`	This parameter refers to the attribute type—for example, integer, string, binary—that validates the mapping rules.
`SrcObjectClass`	If the source of the shared attribute is an LDAP-compliant directory, then this parameter names the object class to which the attribute belongs. If the source of the shared attribute is an Oracle Database repository, then this parameter refers to the table name and is mandatory. For other repositories, this parameter may be ignored.
`DstAttrName`	Optional attribute. If it is not specified, then the `SrcAttrName` is assumed. For LDAP-compliant directories, this parameter refers to the name of the attribute at the destination. For Oracle Database repositories, it refers to the ColumnName in the table specified by the `SrcClassName`. For other repositories, this parameter can be appropriately interpreted.
`DstAttrType`	This parameter refers to the attribute type—for example, integer, string, binary. Note that it is up to you, the administrator, to ensure the compatibility of the source and destination attribute types. Oracle Directory Integration Platform does not ensure this compatibility.
`DstObjectClass`	For LDAP-compliant directories, this parameter refers to the object class to which the attribute belongs, and is optional. For Oracle Database repositories, it refers to the table name, and is mandatory. For other repositories this parameter may be ignored.

Table 6–3 (Cont.) Components in Attribute Rules

Component Name	Description
`AttrMapping Rule`	Optional arithmetic expression with the following operators and functions: Operators: + and | Functions: ■ toUpper(string): Transforms everything to uppercase ■ toLower(string): Transforms everything to lowercase ■ trunc(String,char): Truncates the string at the first occurrence of the character. ■ truncl(String,char): Truncates everything on the left of the string at the first occurrence of the character ■ truncr(String,char): Truncates everything in the string that appears on the right side of the character. ■ bin2b64(byte[]): Transforms the binary value to Base64 ■ b642bin(String): Transforms the base64 encoded value to binary ■ dnconvert(dnvalue): Transforms the DN based on the domain mapping rule. If nothing is specified, then the source attribute value is copied as the value of the destination attribute. Literals can be specified with single quotation marks (') or with double quotation marks ("").

To enter mapping rules in a synchronization profile, edit a file that strictly follows the correct format.

Note: When attributes and object classes are defined in the mapping file, it is assumed that source directories contain the respective attributes and object classes defined in the schema.

If a parent container is selected for synchronization, then all its children that match the mapping rules are likewise synchronized. Child containers cannot be selectively ignored for synchronization.

6.4.2.1 Excluding Attributes

You can insert the `AttributeExclusionList` header in map files and identify attributes to be excluded during bootstrap and synchronization. Attributes listed in the `AttributeExclusionList` will be excluded during bootstrap and synchronization.

The following is an example of the `AttributeExclusionList` header with example attributes to exclude:

```
AttributeExclusionList
facsimileTelephoneNumber
telephonenumber
```

Example 6–3 shows an example map file that includes both the `DomainExclusionList` and `AttributeExclusionList` headers. In this example, the entries under `RDN OU=myou, OU=rfi6894748, DC=imtest, DC=com` and `OU=mynewou, OU=rfi6894748, DC=imtest, DC=com` will be excluded, and all filtered entries will exclude (not contain) the `facsimileTelephoneNumber` and `telephonenumber` attributes.

Example 6–3 Example Map File Using DomainExclusionList and AttributeExclusionList Headers

```
DomainRules
ou=rfi6894748, DC=imtest, DC=com : ou=rfi6894748, cn=users, dc=us, dc=oracle,
dc=com:
DomainExclusionList
OU=myou,OU=rfi6894748,DC=imtest,DC=com
OU=mynewou,OU=rfi6894748,DC=imtest,DC=com
###
AttributeRules
# attribute rule common to all objects
objectguid: :binary: :orclobjectguid:string: :bin2b64(objectguid)
ObjectSID: :binary: :orclObjectSID:string:orclADObject:bin2b64(ObjectSID)
distinguishedName: : : :orclSourceObjectDN: :orclADObject
# USER ENTRY MAPPING RULES
# attribute rule for mapping windows LOGIN id
sAMAccountName,userPrincipalName: : :user:orclSAMAccountName:
:orclADUser:toupper(truncl(userPrincipalName,'@'))+"$"+sAMAccountname
# attribute rule for mapping Active Directory LOGIN id
userPrincipalName: : :user:orclUserPrincipalName: :orclADUser:userPrincipalName
# Map the userprincipalname to the nickname attr by default
userPrincipalName: : :user:uid: :inetorgperson:userPrincipalName
# Map the SamAccountName to the nickname attr if required
# If this rule is enabled, userprincipalname rule needs to be disabled
#sAMAccountName: : :user:uid: :inetorgperson
# Assign the userprincipalname to Kerberaos principalname
userPrincipalName: : :user:krbPrincipalName:
:orcluserv2:trunc(userPrincipalName,'@')+'@'+toupper(truncl(userPrincipalName,'@')
)
# This rule is mapped as SAMAccountName is a mandatory attr on AD
# and sn is mandatory on OID. sn is not mandatory on Active Directory
SAMAccountName: : :user:sn: : person:
# attributes to map to cn - normally this is the given name
cn: : :person:cn: :person:
AttributeExclusionList
facsimileTelephoneNumber
telephonenumber
```

6.4.3 Manually Creating New Mapping Files

Oracle recommends using Oracle Enterprise Manager Fusion Middleware Control to create synchronization mapping rules when you create and configure synchronization profiles. You create mapping rules on the Mapping tab described in Creating Synchronization Profiles on page 7-1. The following information is provided for reference if you must create mapping files manually, that is, not using Oracle Enterprise Manager Fusion Middleware Control.

To create new mapping files manually:

1. Identify the containers of interest for synchronization in the source directory.
2. Identify the destination containers to which the objects in the source containers should be mapped. Be sure that the specified container already exists in the directory.
3. Determine the rule to create a DN of the entry to be created in the destination directory. In LDAP-to-LDAP, mapping is normally one-to-one. In non-LDAP-to-LDAP, a domain DN construct rule is required. For example, in the case of synchronizing from a tagged file or Human Resources agent, the mapping

rule may be in the form `uid=%,dc=mycompany,dc=com`. In this case, the `uid` attribute must be present in all the changes to be applied from Oracle Human Resources. The `uid` attribute must be specified as a required attribute, as specified in step 6.

4. Identify the objects that you want to synchronize among directories—that is, the relevant object classes in the source and destination directories. In general, objects that get synchronized among directories include users, groups, organizational units, organizations, and other resources. Identify the actual object classes used in the directories to identify these objects.
5. Identify the properties of the various objects that you want to synchronize among directories—that is, the attributes in the LDAP context. All the attributes of an object need not be synchronized. The properties of users that you might want to synchronize are `cn`, `sn`, `uid`, and `mail`.
6. Define the mapping rules. Each mapping rule has this format:

   ```
   <srcAttrName>:<ReqdFlag>:<srcAttrType>:<SrcObjectClass>:
   <dstAttrName>:<dstAttrType>:<dstObjectClass>: <Mapping Rule>
   ```

   While defining the mapping rule, ensure the following:

   - Every required attribute has a sequence number. For example, if in step 3 the `uid` attribute is identified as required, then assign a value of `1` in place of `<ReqdFlag>`.
   - Every relevant object class has a schema definition on the destination directory.
   - Every mandatory attribute in a destination object class has a value assigned from the source. This is true for standard object classes also, as the different LDAP implementations may not be completely standards-compliant.

   It is not necessary to assign all attributes belonging to a source object class to a single-destination object class. Different attributes of a source object class can be assigned to different attributes belonging to different destination object classes.

   If an attribute has binary values, then specify it as `binary` in the `<attrtype>` field.

Mapping rules are flexible. They can include both one-to-many and many-to-one mappings.

- One-to-many

  One attribute in a connected directory can map to many attributes in the Oracle back-end directory. For example, suppose an attribute in the connected directory is `Address:123 Main Street/MyTown, MyState 12345`. You can map this attribute in the Oracle back-end directory to both the LDAP attribute `homeAddress` and the LDAP attribute `postalAddress`.

- Many-to-one

  Multiple attributes in a connected directory can map to one attribute in the Oracle back-end directory. For example, suppose that the Oracle Human Resources directory represents Anne Smith by using two attributes: `firstname=Anne` and `lastname=Smith`. You can map these two attributes to one attribute in the Oracle back-end directory: `cn=Anne Smith`. However, in bidirectional synchronization, you cannot then map in reverse. For example, you cannot map `cn=Anne Smith` to many attributes.

See Also: The mapping file examples at the end of this chapter

6.4.4 Supported Attribute Mapping Rules and Examples

The attribute mapping rules supported are:

- Concatenation operator (+): Concatenates two string attributes.

  The mapping rule looks like:

  ```
  Firstname,lastname:  :  :  : givenname:  : inetorgperson: firstname+lastname
  ```

  For example, if the `Firstname` is `John` and `LastName` is `Doe` in the source, then this rule results in the `givenname` attribute in the destination with the value `JohnDoe`.

- OR operator (|): Assigns one of the values of the two string attributes to the destination.

  The mapping rule looks like this:

  ```
  Fistname,lastname : : : :givenname: :inetorgperson: firstname | lastname
  ```

  In this example, `givenname` is assigned the value of `firstname` if it exists. If the `firstname` attribute does not exist, then `givenname` is assigned the value of `lastname`. If both the values are empty, then no value is assigned.

- `bin2b64 ( )`: Stores a binary value of the source directory as a base64 encoded value in the destination directory. Typical usage is as follows:

  ```
  objectguid: : : :binary: :orclobjectguid: orcladuser:bin2b64(objectguid)
  ```

  This is required when you need search on the value of (`objectguid`).

- `tolower()`: Converts the String attribute value to lowercase.

  ```
  firstname: : : :givenname: :inetorgperson: tolower(firstname)
  ```

- `toupper ()`: Converts the String attribute value to uppercase.

  ```
  firstname: : : :givenname: :inetorgperson: toupper(firstname)
  ```

- `trunc(str,char)`: Truncates the string beginning from the first occurrence of the specified `char`.

  ```
  mail : : : : uid : : inetorgperson : trunc(mail,'@')
  ```

  For example, if `mail` is `John.Doe@acme.com` in the source, then this rule results in the `uid` attribute in the destination with the value John.Doe.

- `truncl(str, char)`: Truncates the string up to and including the first occurrence of the specified `char`. For example:

  ```
  mail : : : : uid : : inetorgperson : truncl(mail,'@')
  ```

- `truncr(str, char)`: Truncates everything in the string that appears on the right side of the specified `char`. For example:

  ```
  mail : : : : uid : : inetorgperson : truncr(mail,'@')
  ```

- `dnconvert (str)`: Converts DN type attributes if domain mapping is used.

  This example assumes the following domain mapping rule:

  ```
  DomainRules
  ```

```
cn=srcdomain:cn=dstdomain:
```

For example:

```
uniquemember : : : groupofuniquenames : uniquemember : :groupofuniquenames :
dnconvert(uniquemember)
```

In this example, if `uniquemember` in the source is `cn=testuser1,cn=srcdomain,` then `uniquemember` in the destination becomes `cn=test user1,cn=dstdomain`.

- Literals:

```
Userpassword: : :person: userpassword: :person: 'welcome1'
```

6.4.5 Example: Mapping File for a Tagged-File Interface

Based on the preceding discussions, here is a sample mapping file for importing user entries from the Oracle Human Resources database tables by using the tagged-file interface. Note that the source is a non-LDAP directory. This sample file is supplied during installation, at $*ORACLE_HOME*/ldap/odi/conf/oraclehragent.map.master.

```
DomainRules
NONLDAP:dc=myCompany,dc=com:uid=%dc=myCompany,dc=com
AttributeRules
firstname: : : :cn: :person
email : : : :cn: :person: trunc(email,'@')
email : 1 : :uid: :person:trunc(email,'@')
firstname,lastname: : : :cn: :person: firstname+","+lastname
lastname,firstname: : : :cn: :person: lastname+","+firstname
firstname,lastname: : : :sn: :person: lastname | firstname
EmployeeNumber: : : :employeenumber: :inetOrgperson
EMail: : : :mail: :inetOrgperson
TelephoneNumber1: : : :telephonenumber: :person
TelephoneNumber2: : : :telephonenumber: :person
TelephoneNumber3: : : :telephonenumber: :person
Address1: : : :postaladdress: :person
state: : : :st: :locality
street1: : : :street: :locality
zip: : : :postalcode: :locality
town_or_city: : : :l: :locality
Title: : : :title: :organizationalperson
#Sex: : : :sex: :person
###
```

As described earlier, the mapping file consists of keywords and a set of domain and attribute mapping rule entries. The mapping file in this example contains the domain rule `NONLDAP:dc=myCompany,dc=com:cn=%,dc=myCompany,dc=com`.

- This rule implies that the source domain is NONLDAP—that is, there is no source domain.
- The destination domain (`:dc=myCompany,dc=com`) implies that all the directory entries this profile deals with are in the domain `dc=myCompany,dc=com`. Be sure that the domain exists before you start synchronization.
- The domain mapping rule (`:uid=%,dc=myCompany,dc=com`) implies that the data from the source refers to the entry in the directory with the DN that is constructed using this domain mapping rule. In this case, `uid` must be one of the destination attributes that should always have a non null value. If any data corresponding to an entry to be synchronized has a null value, then the mapping

engine assumes that the entry is not valid and proceeds to the next entry. To identify the entry correctly in the directory, it is also necessary that `uid` is a single value.

- In the case of the tagged file, the source entry does not have an object class to indicate the type of object to which it is synchronizing. Note that the `SrcObjectClass` field is empty.
- Every object whose destination is the Oracle back-end directory must have an object class.
- Note that `email` is specified as a required attribute in the sample mapping file. This is because the `uid` attribute is derived from the `email` attribute. Successful synchronization requires the `email` attribute to be specified in all changes specified in the tagged file as follows:

```
Email : 1 : : :uid : : person : trunc(email,'@')
```

- In some cases, the RDN of the DN needs to be constructed by using the name of a multivalued attribute. For example, to construct an entry with the DN of `cn=%,l=%,dc=myCompany,dc=com`, where `cn` is a multivalued attribute, the DomainMappingRule can be in this form: `rdn,l=%,dc=myCompany,dc=com` where `rdn` is one of the destination attributes having a non null value. A typical mapping file supporting this could have the following form:

```
DomainRules
NONLDAP:dc=us,dc=myCompany,dc=com:rdn,l=%,dc=us,dc=myCompany,dc=com
AttributeRules
firstname: : :cn: :person
email : : : :cn: :person: trunc(email,'@')
email : 1: : :rdn: :person: 'cn='+trunc(email,'@')
firstname,lastname: : : :cn: :person: firstname+","+lastname
lastname,firstname: : : :cn: :person: lastname+","+firstname
firstname,lastname: : : :sn: :person: lastname | firstname
EmployeeNumber: : : :employeenumber: :inetOrgperson
EMail: : : :mail: :inetOrgperson
TelephoneNumber1: : : :telephonenumber: :person
TelephoneNumber2: : : :telephonenumber: :person
TelephoneNumber3: : : :telephonenumber: :person
Address1: : : :postaladdress: :person
Address1: : : :postaladdress: :person
Address1: : : :postaladdress: :person
state: : : :st: :locality
street1: : : :street: :locality
zip: : : :postalcode: :locality
town_or_city: 2 : : :1: :locality
Title: : : :title: :organizationalperson
#Sex: : : :sex: :person
###
```

6.4.6 Example: Mapping Files for an LDIF Interface

Sample integration profiles are created as part of the Oracle Directory Integration Platform installation. The property files used to created the sample integration profiles are located in the $*ORACLE_HOME*`/ldap/odi/samples` directory.

> **Note:** See Section 9.4.2, "Configuring the Mapping File" for a sample import mapping file for a connected Oracle database.

The following is an example of a sample import mapping file:

Sample Import Mapping File

```
DomainRules
dc=mycompany.oid,dc=com:dc=mycompany.iplanet,dc=com
AttributeRules
# Mapping rules to map the domains and containers
o: : :organization: o: :organization
ou: : :organizationalUnit: ou: : organizationalUnit
dc: : :domain:dc: :domain
# Mapping Rules to map users
uid : : :person: uid: :inetOrgperson
sn: : :person:sn: :person
cn: : :person:cn: :person
mail: :inetorgperson: mail: :inetorgperson
employeenumber: :organizationalPerson: employeenumber: :organizationalperson
c: : :country:c: :country
l: : :locality: l: :locality
telephonenumber: :organizationalPerson: telephonenumber: :organizationalperson
userpassword: : :person: userpassword: :person
uid: : :person: orcldefaultProfileGroup: :orclUserV2
# Mapping Rules to map groups
cn: : :groupofuniquenames:cn: :groupofuniquenames
member: : :groupofuniquenames:member: :orclgroup
uniquemember: : :groupofuniquenames:uniquemember: :orclgroup
owner: : :groupofuniquenames:owner: :orclgroup
# userpassword: :base64:userpassword: :binary:
```

Notice, in the preceding example that both the source domain and destination domain are specified in the Domain Mapping rule section. In this example, the source and the destination domains are the same. However, you can specify a different destination domain, provided the container exists in the destination directory.

Also notice, in the preceding example, that the attribute rules are divided into two sections: user attribute mapping rules and group attribute mapping rules. Specifying the object class in a mapping rule helps to uniquely map a specific attribute of an object.

6.4.7 Updating Mapping Rules

You can customize mapping rules by adding new ones, modifying existing ones, or deleting some from the mapping rule set specified in the `orclodipAttributeMappingRules` attribute. In general, to perform any of these operations, you identify the file containing the mapping rules, or store the value of the attribute for a file by using an `ldapsearch` command as described in the documentation for your Oracle back-end directory.

6.4.7.1 Adding an Entry to the Mapping Rules File

To add a new entry to the mapping rules file, edit this file and add a record to it. To do this:

1. Identify the connected directory attribute name and the object class that needs to be mapped to the Oracle back-end directory.
2. Identify the corresponding attribute name in the Oracle back-end directory and the object class to which it needs to be mapped.

3. Generate the mapping rule elements indicating the conversion that needs to be done on the attribute values.
4. Load the attribute mapping rule file to the synchronization profile using the managesyncprofiles command.

   For example, if the e-mail attribute of an entry in the source directory needs to be mapped to the unique identifier of the destination, then it can be:

```
Email:  :  : inetorgperson: uid: : person:
```

6.4.7.2 Modifying an Entry in the Mapping Rules File

After you identify an entry to be modified in the mapping rules file, generate the mapping rule element for the desired conversion of attribute values.

6.4.7.3 Deleting an Entry from the Mapping Rules File

After you identify an entry to be deleted in the mapping rules file, you can either delete the entry from the file or comment it out by putting a number sign (#) in front of it.

See Also:

- "Location and Naming of Files" on page 6-20 for the names of the mapping rule files
- Note 261342.1 *Understanding DIP Mapping Files* in My Oracle Support (formerly MetaLink) at: http://metalink.oracle.com/

6.5 Extending Mappings Using Custom Plug-ins

You can extend mapping functionality using custom plug-ins. The oracle.ldap.odip.util.mapapi.IMapOperation Java interface is defined to support plug-ins for new mapping operations. This topic explains Oracle Directory Integration Platform support for custom plug-ins to extend mapping functionality and contains the following sections:

- Writing Custom Plug-Ins
- Mapping Plug-In Evaluation Constraints
- Adding Mapping Plug-Ins
- Applications of Mapping Plug-Ins
- Example Plug-In Usage

6.5.1 Writing Custom Plug-Ins

To extend mapping functionality using custom plug-ins you must implement the oracle.ldap.odip.util.mapapi.IMapOperation interface, which requires implementing the evaluate method as follows:

```
Vector evaluate(Vector operands);
```

The `operands` argument is a vector. Elements of the `operands` vector can be one of the following, based on the plug-in invocation given in the mapping rule:

- Vector of values (attributes passed as argument for the plug-in)
- String (String literal is passed as argument for the plug-in)

- Character (Character literal)

Return type is a Vector. All elements of this Vector must be Strings or byte arrays. If you want to return a single string, a new vector of size 1 must be created and the string has to be added to it. This restriction is enforced to allow multi-valued attributes.

For example:

```
cn,sn: : :person:description: :person:PLUGIN#MyPlugin(cn, sn, "Mr")
```

The plug-in class MyPlugin should implement Vector evaluate(Vector operands) method. As per the plug-in invocation in the above mapping rule, the following are the elements of operands:

- element1 is a Vector containing all values of cn (Even if cn has only a single value)
- element2 is a Vector containing all values of sn (Even if sn has only a single value)
- element3 is a String literal "Mr"

6.5.2 Mapping Plug-In Evaluation Constraints

- If an attribute has multiple values, the corresponding plug-in will be called only once with all the attribute values stored in a Vector. The plug-in will not be called once per each attribute value.
- Empty String literals (" ") or Character literals (' ') will be ignored.
- You must identify the type of each element in the vector operands of the evaluate() method and process accordingly, as per the plug-in invocation.
- A combination of plug-ins and the existing mapping rule operators or functions is not supported. For example, the following combination is not supported as mapping rule:

  ```
  Plugin#MyPlugin(cn, sn) + givenanme
  toupper(Plugin#(MyPlugin(cn,sn))
  Plugin#TempPlugin1(cn) + Plugin#TempPlugin2(sn)
  ```

- Oracle recommends that Mapping plug-in invocation in different attribute rules follow the same invocation signature. The following example is not recommended and is highly error prone because Myplugin has different invocation signatures:

  ```
  sn: : :person:givenname: :person:PLUGIN#Myplugin(sn,"Mr")
  cn: : :person:description: :person:PLUGIN#Myplugin(cn)
  ```

6.5.3 Adding Mapping Plug-Ins

To add a mapping plug-in to Oracle Directory Integration Platform:

1. If it is running, stop the WebLogic Managed Server hosting Oracle Directory Integration Platform.
2. Copy the mapping plug-in JAR file to the /APP-INF/lib/ directory in the path where the Oracle Directory Integration Platform application was exploded. For example:

   MW_HOME/user_projects/domains/*DOMAIN_NAME*/servers/*MANAGED_SERVER_NAME*/tmp/_WL_user/DIP_*VERSION_NUMBER*/*RANDOM_CHARACTERS*/APP-INF/lib/

3. Start the WebLogic Managed Server hosting Oracle Directory Integration Platform.

6.5.4 Applications of Mapping Plug-Ins

This section describes various applications of Mapping plug-ins, including:

- Support for New Mapping Operations
- Support for Multiple Literal Values

6.5.4.1 Support for New Mapping Operations

Applications can implement their own mapping operations that are not supported internally by the mapping framework.

Support for Conditional Mapping

Conditional Attribute Mapping Support

You can support attribute mapping based on a condition. For example, a mapping rule can be written such that, if the `credential` attribute is present, then `orclisenabled` is set to `ENABLED`, and, if not, `orclisenabled` is set to `DISABLED`. This logic can be supported by implementing a plug-in to assign this value. The mapping rule should be as follows:

```
credential: : :UserType:orclisenabled::orcluserv2:PLUGIN#ConditionalAttrBasedOnPresence(credential)
```

The `PLUGIN#` keyword must be in the attribute mapping rule for any custom plugin (in this case, `ConditionalAttrBasedOnPresence`).

Conditional DN Mapping Support

You can support DN container mapping based on a condition. For example, users must be mapped to container `ou=sales,dc=acme,dc=com` if department is `Sales` and mapped to container `ou=IT,dc=acme,dc=com` if department is `IT`. To support this mapping:

- The DomainRules section can have a **construction rule** like:

  `NONLDAP:dc=acme,dc=com`**`:cn=%,ou=%,dc=acme,dc=com`**

- The AttributeRules section can have a rule with a plug-in operation to map ou as follows:

  ```
  department: : :UserType:ou: :orcluserv2:ConditionalOUMapping(department)
  ```

6.5.4.2 Support for Multiple Literal Values

The current mapping framework only supports specifying a single literal value for an attribute. However, there might be a need to specify more than one literal value when an attribute can have multiple default values. For example, in case of Microsoft Exchange, there is a showInAddressBook attribute which can have more than one value. This can also be implemented using plug-ins.

6.5.5 Example Plug-In Usage

This section provides examples of plug-in usage.

Example 1: Attribute Mapping Rule

```
cn: : :person:initials: :person:PLUGIN#PluginSamp1(cn)
```

Example 1: Corresponding Plug-In Implementation

```
Vector evaluate(Vector operands)
{
Vector all_cnValues = (Vector)operands.get(0);
Vector result =  new Vector();
       ….
       ….
      //All the elements of this result must be strings.
      return result;
}
```

Example 2: Attribute Mapping Rule

```
cn: : :person:givenname: :person:PLUGIN#Myplugin(cn,"Mr")
```

Example 2: Corresponding Plug-In Implementation

```
Vector evaluate(Vector operands)
{
Vector all_cnValues = (Vector)operands.get(0);
String strOperand = (String)operands.get(1);
Vector result =  new Vector();

for(int i=0; i<all_cnValues.size(); i++)
{
String cnValue = (String) all_cnValues.get(i);
String givenNameNewValue = strOperand + cnValue;
result.add(givenNameNewVlaue);
}
      //All the elements of this result must be strings.
      return result;
}
```

Example 3: Attribute Mapping Rule

```
mail: : :inetorgperson:mail: :inetorgperson: Plugin#MyPlugin(mail, '@')
```

Example 3: Corresponding Plug-In Implementation

```
Vector evaluate(Vector operands)
{
        Vector all_mailValues = (Vector) operands.get(0);
        Character charOperand = (Character) operands.get(1);
        char charOperandValue = charOperand.charValue();
Vector result =  new Vector();
       ….
       ….
       ….
       return result;
}
```

Example 4: Attribute Mapping Rule

```
cn,sn,mail: : :inetorgperson:description: :inetorgperson Plugin# MyPlugin(cn, sn,
mail)
```

Example 4: Corresponding Plug-In Implementation

```
Vector evaluate(Vector operands)
{
        Vector all_cnValues = (Vector) operands.get(0);
        Vector all_snValues = (Vector) operands.get(1);
        Vector all_mailValues = (Vector) operands.get(2);

        Vector result = new Vector();
        …
        …
        …

        return result;
}
```

6.6 Configuring Matching Filters

By default, a connector retrieves changes to all objects in the container configured for synchronization. However, you may want to synchronize only certain types of changes, such as changes to just users and groups. While mapping rules allow you to specify how entries are converted from one directory to another, you can also filter objects that are synchronized among directories.

Before changes from a connected directory are imported into the Oracle back-end directory, they can be filtered with the Connected Directory Matching Filter (`orclODIPConDirMatchingFilter`) attribute in the synchronization profile. Similarly, before changes are exported from the Oracle back-end directory to a connected directory, they can be filtered with the OID Matching Filter (`orclODIPOIDMatchingFilter`) attribute.

For both attributes, you can specify a filter for connected directories that either obtain incremental changes through an LDAP search or that store changes in a change log, as described in the following sections:

- Filtering Changes with an LDAP Search
- Filtering Changes from a Change Log

6.6.1 Filtering Changes with an LDAP Search

For connected directories that do not support change logs, the latest footprint of the entries are obtained by performing an LDAP search. Because an LDAP search that is performed with `objectclass=*` will return all entries in a given tree or subtree, to retrieve only the objects of interest for synchronization, you must provide a filter using LDAP filter syntax. For example, you can assign a search filter to the `orclOdipConDirMatchingFilter` attribute. Specify the filter as `searchfilter=`*`LDAP_SEARCH_FILTER`*.

The following example creates an LDAP search filter that retrieves organizational units, groups, and users, but not computers:

```
searchfilter=(|(objectclass=group)(objectclass=organizationalUnit)
(&(objectclass=user)(!(objectclass=computer))))
```

6.6.2 Filtering Changes from a Change Log

For connected directories that store changes in a change log, you can use the following simple operators, which are provided by Oracle Directory Integration Platform, to

specify a matching filter for either the Connected Directory Matching Filter (`orclODIPConDirMatchingFilter`) or the OID Matching Filter (`orclODIPOIDMatchingFilter`):

- = (equal operator)
- ! (not equal operator)

> **Note:** You can use the preceding operators with either an LDAP or non-LDAP directory, provided the directory obtains incremental changes from a change log.
>
> Connected directories that obtain incremental changes through an LDAP search can also use the preceding operators, however, you can only specify a single expression or the search will fail.

Specify the filter as `searchfilter=`*`CHANGELOG_SEARCH_FILTER`*`.`

For example, the following filter prevents syncing if a change is made by profile `imp1` OR profile `imp2`:

```
searchfilter=(|(!(modifiersname=orclodipagentname=imp1,cn=subscriber
profile,cn=changelog subscriber,cn=oracle internet
directory))(!(modifiersname=orclodipagentname=imp2,cn=subscriber
profile,cn=changelog subscriber,cn=oracle internet directory)))
```

For connected directories that store changes in a change log, a matching filter can synchronize changes for only the attributes that appear in the change log. If you include attributes in a matching filter that do not appear in the change log, the search operation will fail. For this reason, matching filters are of limited use for connected directories that store incremental changes in a change log.

6.7 Location and Naming of Files

Table 6–4 lists where to find the various files used during synchronization. By default, when file based interfaces (Tagged/LDIF) are used for synchronization, the files are read from and written to the following locations.

Table 6–4 Location and Names of Files

File	File Name
Import data file	$*ORACLE_HOME*/ldap/odi/data/import/*Profile_Name*.dat
Export data file	$*ORACLE_HOME*/ldap/odi/data/export/*Profile_Name*.dat

For example, the name of the data file of the Oracle Human Resources profile is `oraclehrprofile.dat`.

7

Managing Directory Synchronization Profiles

This chapter explains how to manage directory synchronization profiles. It contains these topics:

- Managing Synchronization Profiles Using Fusion Middleware Control
- Managing Synchronization Profiles Using manageSyncProfiles
- Modifying the Synchronization Status Attributes
- Setting Null Values in Synchronization Profiles

7.1 Managing Synchronization Profiles Using Fusion Middleware Control

This section explains how to create, modify, and delete synchronization profiles by using Oracle Enterprise Manager Fusion Middleware Control. It contains these topics:

- Creating Synchronization Profiles
- Editing Synchronization Profiles
- Enabling and Disabling Synchronization Profiles
- Deleting Synchronization Profiles
- Troubleshooting Synchronization Profiles Using DIP Tester

> **Note:** Users with non-administrator privileges can use Oracle Enterprise Manager Fusion Middleware Control to view information about existing synchronization profiles, but cannot create or edit profiles.

7.1.1 Creating Synchronization Profiles

This section explains how to create synchronization profiles using Oracle Enterprise Manager Fusion Middleware Control. When you create the profile, Oracle recommends using the **Test Connection** function to test the connection to the source host and using the **Validate All Mapping Rules** function to test your mapping rules. If you encounter *error* messages, you must fix the profile configuration or you will not be able to enable the profile and perform synchronization using the profile.

If you create a Synchronization Profile using any of the sample map files included with Oracle Directory Integration Platform, you may encounter various warning messages. The Synchronization Profile will function correctly despite the warnings and you can ignore the warning messages. To avoid the warning messages, edit the default settings

of the map file included with Oracle Directory Integration Platform according to your specific environment, then create the profile.

Perform the following steps to create a synchronization profile using Oracle Enterprise Manager Fusion Middleware Control:

1. Open a Web browser and enter the Oracle Enterprise Manager Fusion Middleware Control URL for your environment. The format of the Oracle Enterprise Manager Fusion Middleware Control URL is: https://*host*:*port*/em.
2. Log in to Oracle Enterprise Manager Fusion Middleware Control.
3. In the navigation panel on the left, click or expand the **Identity and Access** entry and then select the **DIP** component where you want to create the synchronization profile.
4. Click the **DIP Server** menu, point to **Administration**, and then click **Synchronization Profiles**.

   The Manage Synchronization Profiles appears.
5. Click **Create**.

   The Create Synchronization Profile page appears with tabs for the various types of profile settings. The following sections describe the parameters on each tab in the Create Synchronization Profile page.

   After you set values for the parameters, click **OK** on the Create Synchronization Profile page to create the profile. The profile will appear on the Manage Synchronization Profiles page.

General

The General tab contains the following parameters that configure the general settings for the profile:

- **Profile Name**: Specify the name of the connector in ASCII characters only—non-ASCII characters are not supported in the Profile Name. The name you enter is used as the RDN component of the DN for this connector profile. For example, specifying a profile name *MSAccess* creates a connector profile named `orclodipagentname=MSAccess,cn=subscriber profile, cn=changelog subscriber, cn=oracle internet directory`.
- **Profile Status**: Select whether or not to enable or disable the profile.
- **Use DIP-OID as? / Use DIP-OUD as? / Use DIP-ODSEE as?**: This label refers to your installed Oracle directory (either Oracle Internet Directory, Oracle Universal Directory, or Oracle Directory Server Enterprise Edition) that is one end-point for synchronization and provisioning. Select whether your Oracle directory will be used as the source or destination directory. Selecting Source pulls changes from a connected target directory into your Oracle back-end directory. Selecting Destination pushes changes from your Oracle back-end directory into a connected target directory.
- **Type**: Select the type of connected directory from the list.

> **Note:** If you select non-standard LDAP type of profile, such as Database or Custom, the subsequent configuration parameters will vary. For example, if you select Custom from the Type list, you must identify the Java classname and the package, for example: `com.comp.dip.integration.MyListener`

- **Host**: The host where the connected directory is running.
- **Port**: The port where the connected directory is running.
- **SSL Settings**: Specify whether to enable or disable SSL settings. If you enable SSL Settings, the root certificates of the target directory must be in the Oracle Directory Integration Platform keystore to successfully connect or test the connection to the target directory.
- **Database Service ID:** If you selected "Database (JDBC)" from the Type menu, enter the database SID. (Note: Enter the SID, not the service name.)
- **User Name**: Specify the account to be used by the connector agent for accessing the connected directory. For example, if the connected directory is a database, then the account might be Scott. If the connected directory is another LDAP-compliant directory, then the account might be cn=Directory Manager.
- **Password**: Specify the password the connector/agent is to use when accessing the connected directory.
- **Test Connection**: Use the Test Connection function to test the connection to the source host.

Mapping

The Mapping tab allows you to configure Domain and Attribute Mapping Rules, and Domain and Attribute Exclusion Lists for the profile.

Domain Mapping Rules are for the domain or container from which objects are synchronized into the Oracle back-end directory. The Domain Exclusion List identifies domains to be excluded during bootstrap and synchronization.

Attribute Mapping Rules are for attributes of the objects that are being managed. The Attribute Exclusion List identifies attributes to be excluded during bootstrap and synchronization.

To create a mapping rule or exclusion list, click **Create** for the type of mapping rule or exclusion list you want to create, enter values for the parameters, and then click **OK** at the top of the Create Synchronization Profile page.

Note: Use the **Validate All Mapping Rules** button at the top of the Create Synchronization Profile page to test your mapping rules after you create them. If your mapping rules are not valid, you cannot use the profile.

The following is a list and description of the Domain Mapping Rules parameters:

- **DIP-OID Container / DIP-OUD Container / DIP-ODSEE Container**: This label refers to your installed Oracle directory (either Oracle Internet Directory, Oracle Universal Directory, or Oracle Directory Server Enterprise Edition) that is one end-point for synchronization and provisioning. This is the name of the destination container into which the objects are synchronized. Enter a value of NONLDAP if you a synchronizing with a non-LDAP source.
- **Source Container** or **Destination Container**: If you are configuring an import profile, this parameter will be labeled Source Container. If you are configuring an export profile, this parameter will be labeled Destination Container. The parameter identifies the name of the source/destination container from/to which the objects are synchronized. Enter a value of NONLDAP if you a synchronizing with a non-LDAP source.

- **DN Mapping Rule**: The specific mapping rule that determines how entries from the source container are mapped to the destination container.

The following is a list and description of the Domain Exclusion List parameters:

- **Source Container to Exclude**: This parameter appears if you are configuring an import profile. Identify the domains to be excluded during bootstrap and synchronization by entering a value, for example, `OU=myou,OU=test,DC=mycompany,DC=com`, or by clicking Lookup and browsing to the domain, and then clicking **OK** in the Create Domain Exclusion Container dialog box.
- **DIP-OID Container to Exclude / DIP-OUD Container to Exclude / DIP-ODSEE Container to Exclude**: This parameter appears if you are configuring an export profile. Identify the domains to be excluded during bootstrap and synchronization by entering a value, for example, `OU=myou,OU=test,DC=mycompany,DC=com`, or by clicking Lookup and browsing to the domain, and then clicking **OK** in the Create Domain Exclusion Container dialog box.

The following is a list and description of the Attribute Mapping Rules parameters:

- **Source Object Class**: Select the object class in the source directory. This parameter does not apply when synchronizing with a non-LDAP source.
- **Source Attributes**: The source directory attributes to which you want to apply the mapping rule. When synchronizing with LDAP sources, select the **Single Attributes** option and enter the appropriate attributes in the Attributes field. When synchronizing with non-LDAP sources, select the **Multiple Attributes** option and enter the appropriate attributes in the Multivalue Attributes field.
- **Source Attribute Required**: Enable or disable the source attribute requirement.
- **DIP-OID Object Class / DIP-OUD Object Class / DIP-ODSEE Object Class**: Select the destination object type or class. Use the destination object class for LDAP targets.

  Destination Table: If your destination directory type is Database (JDBC), select the destination table.
- **DIP-OID Attribute / DIP-OUD Attribute / DIP-ODSEE Attribute**: Select the destination attribute name to which you want to apply the mapping rule.

  Destination Column: If your destination directory type is Database (JDBC), select the destination column.
- **DIP-OID Attribute Type / DIP-OUD Attribute Type / DIP-ODSEE Attribute Type**: Enter the type of the attribute in the destination directory.
- **Mapping Expression**: Enter the transformation rule that derives the destination attribute value from the source attribute value.

The following is a list and description of the Attribute Exclusion List parameters:

- **ObjectClass**: Select the objectclass that contains the attributes you want to add to the Attribute Exclusion List. After you select an objectclass, its attributes appear in the Multiple Address field.
- **Attributes**: Select the attributes you want to add to the Attribute Exclusion List.

Filtering

The Filtering tab contains the following parameters that configure the filter settings for the profile:

- **Source Matching Filter**: Specify the attribute that uniquely identifies an entry in the connected directory or specify an LDAP search filter for the connected directory in the format searchfilter=*ldap_search_filter*.
- **Destination Matching Rule**: Specify the attribute that uniquely identifies records in the Oracle back-end directory. This attribute is used as a key to synchronize the Oracle back-end directory with the connected directory.
- **Associated Profile**: The Associated Profile filtering setting is used to avoid loop back changes in bi-directional synchronization where changes initiated from one directory return to the same directory. For import profiles, specify the export profile it is associated with in the Associated Profile field. For export profiles, specify the import profile used for synchronizing the data from that directory.

> **Note:** To disassociate a profile, set the Associated Profile setting to **Select One**.

Advanced

The Advanced tab contains the following parameters that configure the advanced settings for the profile:

- **Scheduling Interval (HH:MM:SS)**: Specify the number of hours, minutes, and seconds between synchronization attempts between a connected directory and the Oracle back-end directory.
- **Maximum Number of Retries**: Specify the maximum number of times the synchronization is to be retried before synchronization stops. The default is 5. The first retry takes place one minute after the first failure. The second retry happens two minutes after the second failure, and subsequently the retry takes place *n* minutes after the *n* failure.
- **Log Level**: Specify the logging level for debugging. Selecting the All level logs all information, including entries that are synchronized.
- **Primary Table**: Choose from the list the primary table for this profile.
- **Last Change Number**: Identifies the number of changes that synchronization has been performed for. When you create a synchronization profile, the Last Change Number parameter is locked—you cannot enter a value for it.

  After you create a synchronization profile and attempt to edit it, an additional option named Edit and Persist is available for the Last Change Number parameter. You can edit the value for the Last Change Number parameter if you select (enable) the Edit and Persist option. Enabling the Edit and Persist option causes the Last Change Number to be persisted in the profile. Changes to the Last Change Number will not be persisted if the Edit and Persist option is not enabled.

  > **WARNING: Be aware that if you edit the value for the Last Change Number, setting an incorrect value can cause the profile to stop working or cause erroneous synchronization operations.**

- **Primary Keys**: Specify the primary key(s) for the tables to which you are syncing by selecting the database table name, then entering the primary key column(s). If a

primary key consists of multiple columns, then list each column name separated by a comma. For example: `id,name,dob`. To delete a row, click the red "x" in the row that you want to delete. To add additional primary key entries, click **Add Primary Key**.

- **Table Relations:** Click **Add Table Relation** to define the relationships between the primary table and all of the other tables involved in the profile. In the **Relation Column(s)** box, type the column name that defines the relationship between the Secondary Table and the Primary Table. If you need to specify multiple column names, use a comma separated list, for example: `id,name`.
- **Additional Configuration Parameters**: This section allows you to manage *optional*, advanced configuration parameters. To create an advanced configuration parameter, click **Add** and identify the parameter and its value. The following is a list and description of each advanced configuration parameter:
  - **Check All Entries**: Applicable only for eDirectory and OpenLDAP, it determines how deleted entries in Novell eDirectory or OpenLDAP are synchronized with the Oracle back-end directory. If you assign a value of true to this parameter, the Oracle Directory Integration Platform identifies deleted entries by performing a linear comparison between the entries in the Oracle back-end directory and Novell eDirectory or OpenLDAP. If an entry does not exist in Novell eDirectory or OpenLDAP, the entry is deleted from the Oracle back-end directory. If you assign a value of false to this parameter, deleted entries are synchronized according to the difference between the number of entries in the connected directory and the number of entries in the Oracle back-end directory. If the number of deleted entries is 0 or less than 0, then there are no deleted entries to synchronize. However, if the number of deleted entries is greater than 0, then the Oracle Directory Integration Platform compares each entry in the Oracle back-end directory with Novell eDirectory or OpenLDAP to identify the deleted entries to synchronize. The Oracle Directory Integration Platform continues to compare entries until it locates the same number of deleted entries as the difference between the number of entries in the connected directory and the number of entries in the Oracle back-end directory. For better performance, you should assign a value of false to this parameter.
  - **Unique Attribute**: Applicable only for eDirectory and OpenLDAP, it identifies the unique attribute in Novell eDirectory or OpenLDAP that can be used to search for an entry. You assign to this parameter a value of GUID for Novell eDirectory or entryuuid for OpenLDAP.
  - **Attribute Type**: Applicable only for eDirectory and OpenLDAP, it indicates the type of the UniqueAttribute parameter. You assign to this parameter a value of Binary for Novell eDirectory or nonBinary for OpenLDAP. This parameter is used to obtain the corresponding Oracle back-end directory attribute for the attribute that is defined in the mapping file.
  - **Search Time Delta Size in seconds:** This parameter is applicable only for eDirectory and OpenLDAP, which handle synchronization based on timestamps and do not support changelog. Search Time Delta Size in seconds determines the time interval for processing changes during each synchronization cycle iteration. The default value is 3600. The number of iterations performed during each synchronization cycle depend on the number of pending changes. For example, if the Search Time Delta In Seconds parameter is set to 60 and there are changes pending for about one minute, synchronization will require a single iteration. If changes are pending for three minutes, synchronization will require three iterations.

Notes:

- When the number of changes per minute is small, you will experience better synchronization efficiency by setting Search Time Delta Size in seconds to a higher value.
- Be sure the value you set for the Search Time Delta In Seconds parameter does not exceed the LDAP search limit of the connected directory server. Otherwise, you may receive an error during synchronization and some changes may not be processed.

- **Search Delta Size:** This parameter is applicable when importing changes from directories that support changelog. Search Delta Size determines how many incremental changes are processed during each iteration in a synchronization cycle. The default value is a value of 500. The number of iterations performed during each synchronization cycle depends on the number of pending changes. For example, if the Search Delta Size parameter is assigned a value of 500 and there are 498 pending changes, synchronization will require a single iteration. However, if there are 501 pending changes, synchronization will require two iterations. In some cases, you will experience better synchronization efficiency if you assign a higher value to this parameter. However, be sure that the value you specify does not exceed the LDAP search limit of the connected directory server. Otherwise, you may receive an error during synchronization and some changes may not be processed.

  Note: Be sure to thoroughly analyze and test your deployment when modifying the Search Delta Size parameter, especially if you assign a value higher than 2000.

- **Skip Error To Sync Next Change:** Determines how Oracle Directory Integration Platform handles an error when processing a change during synchronization. By default, Skip Error To Sync Next Change is assigned a value of false, which means that Oracle Directory Integration Platform will continue processing a change until the error is resolved. If you assign a value of true to Skip Error To Sync Next Change, Oracle Directory Integration Platform will skip any changes that cause an error. All failures are recorded in the $*ORACLE_HOME*/ldap/odi/log/profile_name.aud audit log. If you set Skip Error To Sync Next Change to true, be sure to periodically review the audit log for failures.
- **Update Search Count:** Specifies the maximum number of iterations to perform on the connected directory during the synchronization process. The synchronization process stops after the specified number of search has been performed and resumes at the next scheduled interval.
- **Reduce Filter Time In Seconds**: Applicable only for eDirectory and OpenLDAP, it specifies the time difference between a computer that is running the Oracle back-end directory and a computer that is running Novell eDirectory. This parameter is necessary because synchronization between the Oracle back-end directory and Novell eDirectory will not function properly if the time on the Novell eDirectory computer is earlier than the time on the Oracle back-end directory computer. Assign to this parameter a value in seconds that is equal to the time difference between the two computers. The default value is 0.

- **Writer**: Identifies the Writer used by the profile for synchronization. This is a read only value and is used only for information purposes.
- **Reader**: Identifies the Reader used by the profile for synchronization. This is a read only value used only for information purposes.
- **Reconciler**: Do not modify this parameter. It identifies the class used by the profile for reconciliation purposes. The parameter is applicable only for eDirectory and OpenLDAP. This is a read only value used only for information purposes.

7.1.2 Editing Synchronization Profiles

To edit an existing synchronization profile using Oracle Enterprise Manager Fusion Middleware Control:

1. Open a Web browser and enter the Oracle Enterprise Manager Fusion Middleware Control URL for your environment. The format of the Oracle Enterprise Manager Fusion Middleware Control URL is: https://*host*:*port*/em.
2. Log in to Oracle Enterprise Manager Fusion Middleware Control.
3. In the navigation panel on the left, click or expand the **Identity and Access** entry and then select the **DIP** component that contains the profile you want to edit.
4. Click the **DIP Server** menu, point to **Administration**, and then click **Synchronization Profiles**. The Manage Synchronization Profiles appears displaying a list of the existing profiles.
5. Select the profile you want to edit from the list and click **Edit**. The Edit Synchronization Profile screen appears for the profile you want to edit.
6. Edit the profile settings by referring to the "General", "Mapping", "Filtering", and "Advanced" sections in "Creating Synchronization Profiles" on page 7-1 that describe each profile parameter.

   Note: You must edit the settings on the General tab before editing the settings on any other tab.

7. Click **OK** on the Edit Synchronization Profile page to save the updated profile.

7.1.3 Enabling and Disabling Synchronization Profiles

To enable or disable an existing synchronization profile using Oracle Enterprise Manager Fusion Middleware Control:

1. Open a Web browser and enter the Oracle Enterprise Manager Fusion Middleware Control URL for your environment. The format of the Oracle Enterprise Manager Fusion Middleware Control URL is: https://*host*:*port*/em.
2. Log in to Oracle Enterprise Manager Fusion Middleware Control.
3. In the navigation panel on the left, click or expand the **Identity and Access** entry and then select the **DIP** component that contains the profile you want to enable or disable.
4. Click the **DIP Server** menu, point to **Administration**, and then click **Synchronization Profiles**. The Manage Synchronization Profiles appears displaying a list of the existing profiles.
5. Select the profile you want to enable or disable from the list of existing profiles.

Click the **Enable Profile** button to enable the profile.

Click the **Disable Profile** button to enable the profile.

7.1.4 Deleting Synchronization Profiles

Never delete a synchronization profile directly from the Oracle back-end directory! Instead, use Oracle Enterprise Manager Fusion Middleware Control to delete a synchronization profile. If you use the Oracle back-end directory to delete a synchronization profile, you will receive a `PROFILE_ALREADY_REGISTERED` message if you attempt to recreate the profile.

Perform the following steps to delete a synchronization profile using Oracle Enterprise Manager Fusion Middleware Control:

1. Open a Web browser and enter the Oracle Enterprise Manager Fusion Middleware Control URL for your environment. The format of the Oracle Enterprise Manager Fusion Middleware Control URL is: https://*host*:*port*/em.
2. Log in to Oracle Enterprise Manager Fusion Middleware Control.
3. In the navigation panel on the left, click or expand the **Identity and Access** entry and then select the **DIP** component that contains the profile you want to delete.
4. Click the **DIP Server** menu, point to **Administration**, and then click **Synchronization Profiles**. The Manage Synchronization Profiles appears.
5. On the Manage Synchronization Server page, select profile you want to delete and click **Delete**. A window that prompts you to confirm deletion of the connector profile.
6. Click **Yes** to confirm that you want to delete the profile.

7.1.5 Troubleshooting Synchronization Profiles Using DIP Tester

DIP Synchronization Profile Tester (DIP Tester) is a utility that can perform synchronization operations and return detailed log messages generated during the test. Use DIP Tester to test synchronization profiles and verify that the profile configuration and mapping rules are working as expected with your directory data.

DIP Tester only works with LDAP-to-LDAP directory-based sync profiles. It does not work with DB/Custom and other non-directory-based profiles.

You can run DIP Tester either from the Enterprise Manager user interface or from a command-line using WLST.

7.1.5.1 Running DIP Tester From the Enterprise Manager User Interface

DIP Tester can only be used with synchronization profiles that are set to the disabled state.

Note: When disabling a profile there is a delay before the disabled status takes effect. Only after the configured refresh interval has elapsed (an interval of two minutes or more) will the profile become disabled.

To disable a profile click the Disable Profile button on the toolbar.

To Open DIP Tester in Enterprise Manager

1. From the **DIP Server** menu, choose **Administration** > **Synchronization Profiles**.

   The Manage Synchronization Profiles page opens.

2. Click the row of the profile to be tested to select it.

3. Click the arrow to the right of the **DIP Tester** button and choose from the following menu:

   - **Dump Profile** - Opens a pop-up window and displays detailed information about the selected synchronization profile so that a copy of the profile can be added to a support ticket.
   - **Launch Tester** - Opens DIP Tester.

Test Mode

This section describes the Test Mode screen, which is the first screen of the three screen DIP Tester wizard.

DIP Tester Operation Mode - DIP Tester can run in two modes: Basic and Advanced.

- Choose **Basic** mode to quickly run the synchronization task using DIP Tester's preconfigured settings. The synchronization task runs and sync status and log messages from the test are returned in Enterprise Manager.

  Basic mode is useful if you need to quickly retry a failed synchronization. To retry the last failed change, edit the profile and set **Skip Error To Sync Next Change** to false, set **Update Search Count** to 1, and **searchDeltaSize** to 1.

  > **Note:** In the default configuration, Skip Error To Sync Next Change is set to false. Consequently DIP Tester will not move past any failed operations. To have DIP Tester continue on to the next operation upon encountering a failed operation, open the Manage Synchronization Profiles page, click Edit, and on the Advanced tab set Skip Error To Sync Next Change to true.

- Choose **Advanced** mode to specify and configure the source of the test data. There are three ways to specify the test data:
  - Enter a change number to test sync a specific change. Change numbers are supported on Active Directory, Oracle Internet Directory, Oracle Unified Directory, Oracle Directory Server Enterprise Edition, iPlanet, and Tivoli directories.
  - Enter the SourceDN to test sync a specific change. SourceDN is supported on eDirectory and OpenLDAP directories.
  - Enter LDIF data. LDIF data is compatible with all directory sources.

Note: DIP Tester cannot process delete operations for the OpenLDAP or eDirectory LDAP servers.

When processing user or group delete operations on other LDAP servers, LDIF delete files need to include the `dn` and `changetype` attributes.

For example:

```
dn: cn=userToDelete,cn=users,dc=comain
changetype: delete
```

Selected Profile Information - Contains read-only information about the synchronization profile to be tested. Review this information and click **Next** in the top right corner of the screen to go to step two of the wizard, Test Params.

Table 7–1 Synchronization Profile Properties, Basic Properties

Property	Description
Profile Name	The name of the profile to be tested.
Synchronization Mode	Indicates the direction of synchronization. *Import* propagates changes from a connected directory to the Oracle back-end directory. *Export* propagates changes from the Oracle back-end directory to a connected directory.
Profile Status	Indicates that the profile is disabled if the check box is empty.

Table 7–2 Synchronization Profile Properties, Source Details or Destination Details

Property	Description
Type	The directory type supplying records for the import test (for example, OpenLDAP, Microsoft Active Directory, Tivoli Directory Server, and so on).
Host	Name of the computer hosting the source container for the synchronization test.
Port	Configured port number for the synchronization job connection.
SSL Settings	Indicates that SSL is disabled if the check box is empty.
User Name	User account that the profile uses to authenticate to the source directory.

Table 7–3 Synchronization Profile Properties, Advanced

Property	Description
Last Change Number	Identifies the most recent change number in the change log that synchronization has been performed for.
Skip Error To Sync Next Change	If true, specifies that the profile should continue the synchronization job if an error occurs. If false, the sync job is aborted. To change this value, edit the synchronization profile, select the **Advanced** tab, and edit the parameter in the **Additional Configuration Parameters** section.

Table 7–3 (Cont.) Synchronization Profile Properties, Advanced

Property	Description
Update Search Count	Specifies the maximum number of iterations to be performed on the connected directory during the synchronization process. The synchronization process stops after the specified number of searches has been performed. To change this value, edit the synchronization profile, select the Advanced tab, and edit the parameter in the **Additional Configuration Parameters** section.

Test Params

This section describes the Test Params screen, which is the second screen of the three screen DIP Tester wizard.

If running DIP Tester in Advanced mode, select and configure the source of the test data. To test sync a specific change, either enter a change number or enter the SourceDN. Otherwise, enter synchronization instructions using LDIF (Lightweight Directory Interchange Format) statements.

If running DIP Tester in Basic mode, the options are preconfigured. Click **Next** in the top right corner of the screen to go to step three of the wizard, Review Options and Test Output.

View Test Data Optional. This lookup feature is provided so that you can view change log data, source directory data, and destination directory data without leaving DIP Tester. Select one of the following options form the drop-down menu to view the test data inside of DIP Tester.

- **View Change Log Entry** - For directories that support change logs specify the change log and enter a number in the **Change Number** box. The change number is retrieved from the source.

  Note: Failed change numbers cannot be determined automatically. Instead, refer to audit logs for failed change numbers.

- **View Source Directory Entry** - Select to view an existing source directory entry that you can use as a template. This option should be used to view Active Directory source data, including the uSNChanged attribute.

  The **Source Container** values in the drop-down menu are retrieved using the domain mapping rules set in the profile.

  Enter a value for the **Source RDN** box.

- **View Destination Directory Entry** - Select to view an existing destination directory entry that you can use as a template.

  The **Destination Container** values in the drop-down menu are retrieved using the domain mapping rules set in the profile.

  Enter a value for the **Destination RDN** box.

Test Options From the **Test data source** menu, select **Change Number**, **SourceDN**, or **LDIF Data**, then type the change number, SourceDN, or the LDIF commands that you want to test.

Click **Next** to go to step three of the wizard, Review Options and Test Output.

Review Options and Test Output

This section describes the Review Options and Test Output screen, which is the third screen of the three screen DIP Tester wizard.

> **Caution:** A synchronization test is not a simulation. Initiating the test will cause an actual sync operation to take place. Proceed with caution when using DIP Tester in Advance mode.

If running DIP Tester in Advanced mode, use the Review Test Options section to review the change number or LDIF data that you entered on the previous screen. If running DIP Tester in Basic mode, this section uses preconfigured settings.

- Click **Test** to initiate the synchronization test.
- Click **Dump Profile** to write detailed information about the selected synchronization profile to a pop-up window.

> **Note:** If a source directory entry and a destination directory entry are already in sync, DIP Tester will not report that the sync operation did not occur. Instead, DIP Tester simply reports "Test Passed" because the entry and attribute data in both directories match.

The Test Output section displays the following information:

- **Result** - Either **Test Passed** or **Test Failed**.

> **Note:** When using DIP Tester in Basic mode, the **Test Output** section will occasionally report a result of "Test Passed," when, in fact, errors were reported. For this reason you should always check the **Log Messages for DIP Tester** section to verify that no errors are reported there.

- **Source Entry Details / Destination Entry Details** - If the test passed, displays actual directory data test results for both the source and destination directories.
- **Error Message** - If the test failed, displays a message reporting the reason for the test failure.
- **Log Messages for DIP Tester** - Detailed messages generated during the course of the synchronization test.

7.1.5.2 Running DIP Tester From the WLST Command-Line Interface

To run DIP Tester from a command-line, use the `manageSyncProfiles` command and specify the `testProfile` option.

Operation Mode

DIP Tester (`testProfile`) can run in Basic mode or Advanced mode.

- Use Basic mode to quickly run the synchronization task using DIP Tester's preconfigured settings. The synchronization task runs and sync status and log messages from the test are returned to standard out.

Basic mode is useful if you need to quickly retry a failed synchronization. To retry the last failed change, set **Skip Error To Sync Next Change** to false and set **Update Search Count** to 1.

To run DIP Tester in Basic mode, do not include the `-changenumber`, `-sourcedn`, or `-ldiffile` options.

Note: In the default configuration, Skip Error To Sync Next Change is set to false. Consequently DIP Tester will not move past any failed operations. To have DIP Tester continue on to the next operation upon encountering a failed operation, set Skip Error To Sync Next Change to true.

- Use Advanced mode if you need to specify and configure the source of the test data. You can either enter a change number to run a specific change in a change log, enter a SourceDN to test sync a specific change on either an eDirectory or an Open LDAP directory, or enter LDIF data.

  To run DIP Tester in Advanced mode, include either the `-changenumber`, the `-sourcedn`, or the `-ldiffile` options when running DIP Tester.

  Note: DIP Tester cannot process delete operations for the OpenLDAP or eDirectory LDAP servers.

  When processing user or group delete operations on other LDAP servers, LDIF delete files need to include the `dn` and `changetype` attributes.

  For example:

  ```
  dn: cn=userToDelete,cn=users,dc=comain
  changetype: delete
  ```

Refer to the following DIP Tester Command-Line Examples section for more information.

Note: If a source directory entry and a destination directory entry are already in sync, DIP Tester will not report that the sync operation did not occur. Instead, DIP Tester simply reports "Test Passed" because the entry and attribute data in both directories match.

Syntax and Arguments for testProfile

`manageSyncProfiles testProfile -h` *`hostName`* `-p` *`port`* `-D` *`wlsuser`* `-pf` *`profileName`* `[-changenumber` *`number`* `| -ldiffile` *`file`* `| -sourcedn` *`sourcedn`*`] [-ssl -keyStorePath` *`path`* `-keyStoreType` *`type`*`] [-help]`

-h | -host
Oracle WebLogic Server where Oracle Directory Integration Platform is deployed.

-p | -port
Listening port of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed.

- D | -wlsuser
Login ID to connect to the server.

-pf | -profile
The profile name.

-changenumber
The change log number representing the change that needs to be synchronized from directories like Oracle Internet Directory, iPlanet, Tivoli and Active Directory to the destination.

-sourcedn
Distinguished name of the source entry that needs to be synchronized from directories like eDirectory and OpenLDAP to the destination.

-ldiffile
A file containing LDIF data that will be applied to the source and then synchronized to the destination.

- ssl
Executes the command using SSL.

- keyStorePath
Location of the trust keystore.

- keyStoreType
Keystore type. If unspecified, defaults to `jks`.

- help
Provides help for the testProfile command.

Note: Passwords cannot be passed as command parameters. Instead, the system will prompt you for passwords as needed.

DIP Tester Command-Line Examples

DIP Tester Basic Mode

```
manageSyncProfiles testProfile -h <hostName> -p <port> -D
<wlsuser> -pf <profileName>
```

DIP Tester Advanced Mode, Change Number Specified

```
manageSyncProfiles testProfile -h <hostName> -p <port> -D
<wlsuser> -pf <profileName> [-changenumber <number>]
```

DIP Tester Advanced Mode, SourceDN Specified

```
manageSyncProfiles testProfile -h <hostName> -p <port> -D
<wlsuser> -pf <profileName> [-sourcedn <sourcedn>]
```

DIP Tester Advanced Mode, LDIF File Specified

```
manageSyncProfiles testProfile -h <hostName> -p <port> -D
<wlsuser> -pf <profileName> [-ldiffile <file>]
```

7.2 Managing Synchronization Profiles Using manageSyncProfiles

Use the `manageSyncProfiles` utility to create and manage synchronization profiles from a command line. The `manageSyncProfiles` utility is located in the *ORACLE_HOME*`/bin` directory.

Notes:

- Best security practice is to provide a password only in response to a prompt from the command.
- You must set the `WLS_HOME` and `ORACLE_HOME` environment variables before executing any of the Oracle Directory Integration Platform commands
- The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. Refer to the *Configuring SSL* chapter in *Oracle Fusion Middleware Securing Oracle WebLogic Server* for more information.

This topic contains the following sections:

7.2.1 Syntax for manageSyncProfiles

manageSyncProfiles

```
manageSyncProfiles {activate | deactivate | copy | deregister | get | isexists |
update | testProfile | validateProfile | validateMapRules | register |
updatechgnum | associateProfile | dissociateProfile | getAllAssociatedProfiles |
getAssociatedProfile | list } -h HOST -p PORT -D wlsuser [-ssl -keystorePath
PATH_TO_KEYSTORE -keystoreType TYPE] [-profile] [-newProfile]
[-associateProfile][-file] [-params 'prop1 val1 prop2 val2 ...']
[-conDirHost] [-conDirPort] [-conDirBindDn] [-mode] [-conDirType] [-conDirSSL]
[-profileStatus] [-help]
```

7.2.2 Arguments for manageSyncProfiles

Operations

activate

Changes the state of the profile identified by -profile to ENABLE.

deactivate

Changes the state of the profile identified by -profile to DISABLE.

copy

Copies an existing profile *profile* to profile *newProfile.*

deregister

Deletes an existing profile from the Oracle back-end directory.

get

Gets the profile details from the Oracle back-end directory.

isexists

Checks if the profile *profile* exists in the Oracle back-end directory.

update

Modifies the profile properties that are identified by command arguments.

testProfile

Changes the state of a disabled profile *profile* to TEST and schedules the profile for testing to ensure the profile will successfully perform synchronization. After executing the manageSyncProfiles command with the testProfile operation, the results of the test are available in the following log file, where *WL_DOMAIN_HOME* represents the Oracle WebLogic Server Domain home and *ORACLE_WEBLOGIC_MANAGEDSERVER_NAME* represents the name of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed:

WL_DOMAIN_HOME/servers/*ORACLE_WEBLOGIC_MANAGED_SERVER_NAME*/logs/*ORACLE_WEBLOGIC_MANAGED_SERVER_NAME*.log

> **Note:** The testProfile operation cannot schedule profiles that are in ENABLE state for testing.

validateProfile

Validates the syntax of the values in the specified profile for correctness.

validateMapRules

Validates the map rules provided.

register

Creates a new profile in the Oracle back-end directory.

updatechgnum

Updates the last applied change number in the profile to latest.

associateProfile

Associates *associateProfileName* with *profileName*. This is helpful during bidirectional synchronization between directories. You can specify a profile as an associated profile of different profile to help prevent information backflow.

dissociateProfile

Dissociates an associated profile to *profileName*.

getAllAssociatedProfiles

Returns a list of all profiles whose `orclodipassociatedprofile` attribute is set to the profile you identify using -pf. For example, if you use `getAllAssociatedProfiles` with `-pf test`, `getAllAssociatedProfiles` returns a list of all profiles that have their `orclodipassociatedprofile` attribute set to test.

This is useful when you want to delete a profile. You can use it to get a list of all associations you must disassociate before you can delete the profile.

getAssociatedProfile

Returns the value of the `orclodipassociatedprofile` attribute for the profile you identify using -pf.

list

Displays all profiles registered in the Oracle back-end directory.

Options

-h | host

Oracle WebLogic Managed Server host where Oracle Directory Integration Platform is deployed.

-p | -port

Listening port of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed.

-D | wlsuser

Oracle WebLogic Server login ID

> **Note:** You will be prompted for the Oracle WebLogic Server login password. You cannot provide the password as a command-line argument. Best security practice is to provide a password only in response to a prompt from the command. If you must execute `manageSyncProfiles` from a script, you can redirect input from a file containing the Oracle WebLogic Server login password. Use file permissions to protect the file and delete it when it is no longer necessary. If you must provide more than one password to `manageSyncProfiles`, put each on a separate line in the file, in the following order: connected directory bind DN password, then Oracle WebLogic Server login password.

-ssl

Executes the command in SSL mode.

> **Note:** The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. Refer to the *Configuring SSL* chapter in *Oracle Fusion Middleware Securing Oracle WebLogic Server* for more information.

-keystorePath

The full path to the keystore.

-keystoreType

The type of the keystore identified by `-keystorePath`. For example: `-keystorePath jks` or `-keystorePath PKCS12`

-pf | -profile

The name of the synchronization profile to use when performing the operation.

-newpf | -newProfile

The name of the new profile which will be a copy of *profile*.

-assopf

The name of the profile that will be associated with *profile*.

-f | -file

The full path and file name of the profile properties file containing the properties.

> **See:** Appendix B, "Example Properties File for Synchronization Profiles" for an example of a profile properties file.

-params

A value is of the form `prop1 val1 prop2 val2 ...` where `prop` is the name of a profile property and `val` is the new value for that property. This keyword is used only for modification of a profile. You can specify as many key values as required. Refer to Appendix B, "Example Properties File for Synchronization Profiles" to see the names of the profile properties that can be identified using `prop1`, `prop2`, and so on.

-conDirHost

Host where connected directory server is running.

-conDirPort

Port at which connected directory server listens.

-conDirBindDn

Connected directory server bind DN.

Note: You will be prompted for the connected directory bind DN password. You cannot provide the password as a command-line argument. Best security practice is to provide a password only in response to a prompt from the command. If you must execute `manageSyncProfiles` from a script, you can redirect input from a file containing the connected directory bind DN password. Use file permissions to protect the file and delete it when it is no longer necessary. If you must provide more than one password to `manageSyncProfiles`, put each on a separate line in the file, in the following order: connected directory bind DN password, then Oracle WebLogic Server login password.

-mode

Synchronization mode to be used: `import` or `export`

-conDirType

Connected directory type. If using Oracle Internet Directory as your Oracle back-end directory, supported values are `ActiveDirectory`, `EDirectory`, `iPlanet`, `OpenLDAP`, `ADAM`, `Tivoli`, `OID`, and `ExchangeServer2003`. If using Oracle Unified Directory or Oracle Directory Server Enterprise Edition as your back-end directory, `IPLANET` is the only supported value.

-conDirSSL

SSL mode value used to connect connected directory server.

-prfSt | -profileStatus

Displays status for the profile. Used only with the `list` operation.

-help

Provides command usage help.

7.2.3 Tasks and Examples for manageSyncProfiles

```
manageSyncProfiles register -h myhost.mycompany.com -p 7005 -D login_ID \
  -f /opt/ldap/odip/iPlImport.profile

manageSyncProfiles deregister -h myhost.mycompany.com -p 7005 \
   -D login_ID -pf myProfile

manageSyncProfiles updatechgnum -h myhost.mycompany.com -p 7005 \
   -D login_ID -pf myProfile

manageSyncProfiles activate -h myhost.mycompany.com -p 7005 \
   -D login_ID -pf myProfile

manageSyncProfiles deactivate -h myhost.mycompany.com -p 7005 \
   -D login_ID -pf myProfile

manageSyncProfiles get -h myhost.mycompany.com -p 7005 \
   -D login_ID -pf myProfile

manageSyncProfiles testProfile -h myhost.mycompany.com -p 7005 \
   -D login_ID -pf myProfile
```

```
manageSyncProfiles associateprofile -h myhost.mycompany.com -p 7005 \
   -D login_ID -pf myProfile -assopf myProfile1

 manageSyncProfiles dissociateprofile -h myhost.mycompany.com -p 7005 \
   -D login_ID -pf myProfile

 manageSyncProfiles getAllAssociatedProfiles -h myhost.mycompany.com -p 7005 \
   -D login_ID -pf myProfile

 manageSyncProfiles getAssociatedProfile -h myhost.mycompany.com -p 7005 \
   -D login_ID -pf myProfile

 manageSyncProfiles update -h myhost.mycompany.com -p 7005 \
   -D login_ID -pf myProfile -f /opt/ldap/odip/iPlImport.profile

 manageSyncProfiles validateMapRules -h myhost.mycompany.com -p 7005 \
   -D login_ID -f /opt/ldap/odip/iPlImport.map -conDirHost server.example.com \
   -conDirPort 8000 -conDirBindDn administrator@idm2003.net -mode IMPORT \
   -conDirType IPLANET

 manageSyncProfiles isexists -h myhost.mycompany.com -p 7005 -D login_ID \
   -pf myProfile

 manageSyncProfiles copy -h myhost.mycompany.com -p 7005 -D login_ID \
   -pf myProfile -newpf yourProfile

manageSyncProfiles list -h myhost.mycompany.com -p 7005 -D login_ID -profileStatus
```

7.3 Modifying the Synchronization Status Attributes

During the synchronization process, the server constantly updates the `orcllastappliedchangenumber` synchronization status attribute. Oracle recommends that you do not change the synchronization status attributes. However, there may be cases when you need to update the `orcllastappliedchangenumber` attribute. For example, you may need to reapply some changes or skip synchronization of certain entries.

You can change the `orcllastappliedchangenumber` attribute using Oracle Enterprise Manager Fusion Middleware Control or the `manageSyncProfiles` command and the `updatechgnum` argument.

To change the `orcllastappliedchangenumber` attribute using Oracle Enterprise Manager Fusion Middleware Control, perform the steps in "Editing Synchronization Profiles" on page 7-8, and set the Last Change Number setting on the Advanced tab.

To change the `orcllastappliedchangenumber` attribute using the `manageSyncProfiles` command and the `updatechgnum` argument, refer to "Managing Synchronization Profiles Using manageSyncProfiles" on page 7-16.

7.4 Setting Null Values in Synchronization Profiles

To set a profile property value to null (that is, blank or empty) when manually editing a profile, use a null string, for example: `''`. Using a comment (or hash character, `#`) on the property's line indicates only that the line will not be read, it does not set the property's value to null.

8

Bootstrapping a Directory in Oracle Directory Integration Platform

This chapter discusses directory bootstrapping, which refers to the initial migration of data between a connected directory and the Oracle back-end directory. Because the synchronization process can handle the migration of data between a connected directory and the Oracle back-end directory, you are not required to perform directory bootstrapping. However, relying on the synchronization process to perform the initial migration can be a time-consuming process, especially for large amounts of data. For this reason, you should perform directory bootstrapping when you first deploy Oracle Directory Integration Platform.

This chapter contains these topics:

> **See Also:** If using Oracle Internet Directory as your back-end directory, see the chapter on data migration from other directories and data repositories in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

8.1 Directory Bootstrapping Using syncProfileBootstrap

Use the `syncProfileBootstrap` utility, located in the *ORACLE_HOME*`/bin` directory, to bootstrap between a connected directory and the Oracle back-end directory.

Notes:

- The `syncProfileBootstrap` command enables you to bootstrap using either a parameter file or a completely configured integration profile. This topic discusses both approaches.
- To bootstrap between a connected Oracle Database and the Oracle back-end directory, configure the export profile `dbexport.cfg` and bootstrap with that profile. See Section 9.4.1, "Configuring the Additional Configuration Information File" for more information.
- Best security practice is to provide a password only in response to a prompt from the command.
- You must set the `WLS_HOME` and `ORACLE_HOME` environment variables before executing any of the Oracle Directory Integration Platform commands
- The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. Refer to the *Configuring SSL* chapter in *Oracle Fusion Middleware Securing Oracle WebLogic Server* for more information.

This topic includes the following sections:

8.1.1 Syntax for syncProfileBootstrap

syncProfileBootstrap

```
syncProfileBootstrap -h HOST -p PORT -D wlsuser {-file FILENAME |-profile
-PROFILE_NAME} [-ssl -keystorePath PATH_TO_KEYSTORE -keystoreType TYPE]
[-loadParallelism INTEGER] [-loadRetry INTEGER][-help]
```

8.1.2 Arguments for syncProfileBootstrap

-h | -host

Oracle WebLogic Server host where Oracle Directory Integration Platform is deployed.

-p | -port

Listening port of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed.

-D | wlsuser

Oracle WebLogic Server login ID

Note: You will be prompted for the Oracle WebLogic Server login password. You cannot provide the password as a command-line argument. Best security practice is to provide a password only in response to a prompt from the command. If you must execute `syncProfileBootstrap` from a script, you can redirect input from a file containing the Oracle WebLogic Server login password. Use file permissions to protect the file and delete it when it is no longer necessary.

-f | -file

Bootstrap properties file.

-pf | -profile

The name of the synchronization profile to use when performing the operation.

-ssl

Executes the command in SSL mode.

Note: The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. Refer to the *Configuring SSL* chapter in *Oracle Fusion Middleware Securing Oracle WebLogic Server* for more information.

-keystorePath

The full path to the keystore.

-keystoreType

The type of the keystore identified by `-keystorePath`. For example: `-keystorePath jks` or `-keystorePath PKCS12`

-lp | -loadParallelism

Indicates that loading to the Oracle back-end directory is to take place in parallel by using multiple threads. For example, `-loadparallelism 5` means that 5 threads are to be created, each of which tries to load the entries in parallel to the Oracle back-end directory.

-lr | -loadRetry

The number of times the retry should be made (when the load to the destination fails) before marking the entry as bad entry.

-help

Provides command usage help.

8.1.3 Tasks and Examples for syncProfileBootstrap

```
manageSyncProfileBootstrap -h myhost.mycompany.com -p 7005 -D login_ID \
  -pf myProfile -lp 5

manageSyncProfileBootstrap -h myhost.mycompany.com -p 7005 -D login_ID \
```

```
-f /opt/ldap/odip/bootstrap.properties -lr 3
```

8.1.4 Recommended Bootstrapping Methodology

If the source directory from which you are loading data contains a large number of entries, the quickest and easiest method to bootstrap the target directory is by using an LDIF file. Bootstrapping with an integration profile is not recommended in this case because connection errors may occur when reading and writing between the source and target directories. Using an LDIF file is also recommended if the DNs contain special characters, which may not be escaped properly when bootstrapping with an integration profile.

8.1.5 Bootstrapping Using a Parameter File

The parameters in this file specify:

- Source and destination interface types (LDIF and LDAP)
- Connection details and credentials (valid only for LDAP)
- Mapping rules

You can bootstrap using an LDIF file by using directory-dependent tools to read from the source directory.

During installation, the following sample parameter files are copied to the $*ORACLE_HOME*`/ldap/odi/conf/` directory:

- `Ldp2ldp.properties`
- `Ldp2ldf.properties`
- `Ldf2ldp.properties`
- `Ldf2ldf.properties`

The preceding files describe the significance of each of the parameters in bootstrapping. When you run the tools for bootstrapping, be sure that the `ORACLE_HOME` and `NLS_LANG` settings are correct.

Bootstrapping can be performed between services with or without one or more intermediate files. However, for large directories, an intermediate LDIF file is required.

8.1.5.1 Bootstrapping Without Using an LDIF File

Oracle recommends this method for smaller directories where the entries are:

- Relatively few in number
- In a flat structure
- Not interdependent—that is, the creation of one entry does not depend on the existence of another as, for example, when the creation of a group entry depends on the existence of user member entries

To use this method:

1. Create the mapping file with appropriate mapping rules. The mapping file is one of the properties in the bootstrap file. Be sure that it is compatible with the mapping rules defined for synchronization.
2. Create the parameter file with the required details specifying the source as LDAP and the destination type as LDIF. A sample parameter file,

`ldp2ldf.properties`, is available in $*ORACLE_HOME*`/ldap/odi/samples`. Make sure that binary attributes are specified as binary in the `SrcAttrType` field.

3. Execute the `syncProfileBootstrap` command with a configuration file that contains:
   - The source is specified as an LDAP directory.
   - The destination type is specified as an LDIF.
4. Check the *NAME_OF_MANAGED_SERVER*`-diagnostic.log` file for any errors. This file can be found in the following location:

   MW_HOME`/user_projects/domains/`*DOMAIN_NAME*`/servers/`*NAME_OF_MANAGED_SERVER*`/logs/`*NAME_OF_MANAGED_SERVER*`-diagnostic.log`
5. Use `bulkload.sh` or the `ldapadd` command to upload the data to the Oracle back-end directory.
6. To continue synchronization, use the `updatechgnum` operation of the `manageSyncProfiles` command to update the last change number, as follows:

   ```
   manageSyncProfiles updatechgnum -h HOST -p PORT -D wlsuser \
   -profile my_Import_Profile
   ```

8.1.5.2 Bootstrapping Using an LDIF File

This section describes the following two ways to bootstrap a directory by using an LDIF file:

- Bootstrapping from an LDIF File Using Directory-Dependent Tools to Read the Source Directory
- Bootstrapping from an LDIF File Using the syncProfileBootstrap Command to Load Data into the Back-end Directory

Bootstrapping from an LDIF File Using Directory-Dependent Tools to Read the Source Directory

Oracle recommends that you use this method for large directories. To use this method:

1. Download the data from the directory to an LDIF file. The tool you use depends on the directory from which you are loading the data. If you are bootstrapping from a Microsoft Active Directory, then use the `ldifde` command to load the data. Be sure to load all the required attributes for each entry.
2. Create the mapping file with appropriate mapping rules. When you want to do further synchronization, be sure that the mapping file is the same as the one used for synchronization.
3. Create the parameter file with source and destination as LDIF and other details. A sample parameter file is available in:

   $*ORACLE_HOME*`/ldap/odi/conf/ldf2ldf.properties`.
4. Use the syncProfileBootstrap command with a parameter file in which the source is specified as LDIF and the destination type is specified as LDIF. This converts the source data and creates a new LDIF as required by Oracle Internet Directory. Run the syncProfileBootstrap command as follows:

   ```
   syncProfileBootstrap -profile profile_name -loadParallelism threads -loadRetry
   retries
   ```

5. Check the *NAME_OF_MANAGED_SERVER*-diagnostic.log file for any errors. This file can be found in the following location:

   MW_HOME/user_projects/domains/*DOMAIN_NAME*/servers/*NAME_OF_MANAGED_SERVER*/logs/*NAME_OF_MANAGED_SERVER*-diagnostic.log

6. Use bulkload.sh or the ldapadd command to upload the data to the Oracle back-end directory.

7. To continue synchronization, use the updatechgnum operation of the manageSyncProfiles command to update the last change number, as follows:

   ```
   manageSyncProfiles updatechgnum -h HOST -p PORT -D wlsuser \
   -profile my_Import_Profile
   ```

Bootstrapping from an LDIF File Using the syncProfileBootstrap Command to Load Data into the Back-end Directory

To use this method:

1. Download the data from the directory to an LDIF file. The tool you use depends on the directory from which you are loading the data. If you are bootstrapping from a Microsoft Active Directory, then use the ldifde command to load the data. Be sure to load all the required attributes for each entry.

2. Prepare the mapping file with appropriate mapping rules. When you want to do further synchronization, be sure that the mapping file is the same as the one used for synchronization.

3. Create the properties file with the source specified as LDIF and the destination specified as LDAP.

4. Use the syncProfileBootstrap command with a parameter file in which the source is specified as the LDIF file, the destination type is specified as LDAP, and the destination specified as Oracle Internet Directory. This converts the source data and creates entries in Oracle Internet Directory as required. A sample properties file, ldf2ldp.properties, is available in $*ORACLE_HOME*/ldap/odi/samples.

5. Check the *NAME_OF_MANAGED_SERVER*-diagnostic.log file for any errors. This file can be found in the following location:

   MW_HOME/user_projects/domains/*DOMAIN_NAME*/servers/*NAME_OF_MANAGED_SERVER*/logs/*NAME_OF_MANAGED_SERVER*-diagnostic.log

6. To continue synchronization, use the updatechgnum operation of the manageSyncProfiles command to update the last change number, as follows:

   ```
   manageSyncProfiles updatechgnum -h HOST -p PORT -D wlsuser \
   -profile my_Import_Profile
   ```

8.1.6 Bootstrapping Directly Using the Default Integration Profile

Bootstrapping relies on an existing integration profile configured for synchronization. This configuration information is used to connect to the other directory.

While using this method, put the source directory in read-only mode.

If the profile is an import profile, then footprints of the required objects in the connected directory are created in the Oracle back-end directory. If the profile is an

export profile, then footprints of the required objects from the Oracle back-end directory are created in the connected directory.

While creating these entries, the distinguished name and object-level mappings as specified in the integration profile are used. If there is a failure uploading the entries, then the information is logged in the *NAME_OF_MANAGED_SERVER*`-diagnostic.log` file located in the *MW_HOME*`/user_projects/domains/`*DOMAIN_NAME*`/servers/`*NAME_OF_MANAGED_SERVER*`/logs` directory.

For example, for bootstrapping from Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server) to Oracle Internet Directory, you would do the following:

1. Customize the default integration profile `iPlanetImport`, which is created as part of the installation by following the instructions in "Configuring Advanced Integration with Oracle Directory Server Enterprise Edition" on page 20-2.
2. Enter the following command:

   ```
   syncProfileBootstrap -h HOST -p PORT -D wlsuser -profile iPlanetImport
   -loadParallelism 5 -loadRetry 3
   ```

3. Check the *NAME_OF_MANAGED_SERVER*`-diagnostic.log` file for any errors. This file can be found in the following location:

   MW_HOME`/user_projects/domains/`*DOMAIN_NAME*`/servers/`*NAME_OF_MANAGED_SERVER*`/logs/`*NAME_OF_MANAGED_SERVER*`-diagnostic.log`

If you use the syncProfileBootstrap command, following the bootstrapping process the `lastchangenumber` attribute is initialized for further synchronization.

8.2 Bootstrapping in SSL Mode

You can use either a parameter file or an integration profile to bootstrap in SSL mode. When you bootstrap in SSL mode, either the Oracle back-end directory, the connected directory, or both the Oracle back-end directory and the connected directory can be running SSL mode.

To bootstrap in SSL mode from a parameter file, you must assign values of either true or false to the `odip.bootstrap.srcsslmode` and `odip.bootstrap.destsslmode` arguments in the parameter file.

When you bootstrap from an integration profile, the value assigned to the default integration profile's `odip.profile.condirurl` is used to establish an SSL connection to the connected directory.

8.2.1 Adding a Trusted Certificate to the DIP Keystore

When bootstrapping in SSL mode, Directory Integration Platform needs to have the trusted certificate of the third party directory in its keystore. DIP will connect to the third party directory using SSL Server-Auth mode.

8.2.1.1 To Add a Trusted Certificate to the DIP Keystore

Complete the following before starting the bootstrap in SSL mode.

1. Create a new Java Key Store using the `keytool` command in some physical location and add the third party directory trusted certificate into this keystore.

```
keytool -importcert -noprompt -trustcacerts -alias <ALIAS_
NAME> -file <PATH_TO_CERTIFICATE_FILE> -keystore <PHYSICAL_
LOCATION_OF_KEYSTORE> -storepass <KEYSTORE_PASSWORD>
```

2. Configure the Java Key Store (JKS) location (created in the previous step) in the Directory Integration Platform application.

   In the following command, WLS stands for WebLogic Server.

   ```
   $OH/bin/manageDIPServerConfig set -attr keystorelocation -val
   <FULL_PATH_TO_KEYSTORE> -h <WLS_HOST> -p <WLS_MANAGED_
   SERVER_PORT> -wlsuser <WLS_USER>
   ```

3. Create a CSF (Credential Store Framework) password credential so that DIP can read the password from CSF and open the keystore for validating the certificates.

   a. Run the following command:

      ```
      $MW_HOME/oracle_common/common/bin/wlst.sh
      ```

   b. Run the following command:

      ```
      connect('%WLSUSER%','%WLSPWD%',
      't3://%HOST%:%ADMINSERVER_PORT%')
      ```

   c. Run the following WLST command to create a credential:

      ```
      createCred(map="dip", key="jksKey",
      user="cn=odisrv,cn=Registered Instances,cn=Directory
      Integration Platform,cn=products,cn=oraclecontext",
      password="<JKS_PASSWORD>", desc="DIP SSL JKS")
      ```

9

Synchronizing with Tables in Oracle Database

You can use either the Oracle Enterprise Manager user interface or the WLST command-line utility to configure Oracle Database import and export profiles.

- See Section 7.1, "Managing Synchronization Profiles Using Fusion Middleware Control," for help creating and managing profiles using the user interface.
- See Section 7.2, "Managing Synchronization Profiles Using manageSyncProfiles," for help creating and managing profiles using the WLST `manageSyncProfiles` command utility.

This chapter describes the "DBReader" configuration files that you can modify from the command line to synchronize data stored in Oracle Database tables with the Oracle back-end directory. The synchronization can be either incremental—for example, one database table row at a time—or all the database tables at once.

This chapter also describes the "dbexport" configuration files that you can modify if you want to create from the command line an export profile that will synchronize an Oracle back-end directory with an Oracle Database.

Note: Multi-valued attribute synchronization from the database to the Oracle back-end directory is not supported.

The Oracle Directory Integration Platform application does not support database delete operations. You can, however, create an outside trigger that will directly delete entries in the Oracle back-end directory whenever a row is deleted in the database. Information about how to create such a trigger is outside of the scope of this documentation.

This chapter contains these topics:

Note: Before reading this chapter, be sure to familiarize yourself with the introductory chapters about Oracle Directory Integration Platform—specifically:

- Chapter 1, "Introduction to Oracle Identity Management Integration"
- Chapter 5, "Understanding the Oracle Directory Synchronization Service"

9.1 Preparing the Additional Configuration Information File

To create an *import* profile that synchronizes data in Oracle Database with the Oracle back-end directory, open the sample file `DBReader.cfg.master` in the $*ORACLE_HOME*`/ldap/odi/conf` directory, and edit it to your specifications.

To create an *export* profile that synchronizes data in the Oracle back-end directory with Oracle Database, open the sample file `dbexport.cfg.master` in the $*ORACLE_HOME*`/ldap/odi/conf` directory, and edit it to your specifications.

Understanding the Format of the Additional Configuration Information File

It is very important to follow the correct format of this file. The various sections are divided using TAG names. Every TAG section has a list of parameters and their respective values. The general layout is as follows:

```
[TAG]
PARAMETER1: value
PARAMETER2: value

[TAG]
PARAMETER1: value
PARAMETER2: value\
VALUE continuation\
value continuation\
end of value continuation

[TAG]
PARAMETER1: value
PARAMETER2: value\
end of value continuation
```

Understanding the DBReader.cfg.master Configuration File

During synchronization from Oracle database to the Oracle back-end directory, the `DBReader.cfg.master` file governs the retrieval of data from the database. It provides the Oracle Directory Integration Platform with the following information:

- The `SELECT` statement to execute
- Either the attributes or the database columns to be used in incremental synchronization. Generally, this is either an attribute that contains a timestamp or a change sequence number that the next SQL statement should use to retrieve incremental data.

The `DBReader.cfg.master` file looks like this:

```
[DBQUERY]
SELECT: SELECT\
EMPNO EmpNum,\
ENAME,\
```

```
REPLACE(EMAIL),'@ACME.COM','') UID,\
EMAIL,\
TELEPHONE,\
TO_CHAR(LAST_UPDATE_DATE,'YYYYMMDDHH24MISS') Modified_Date\
FROM\
EMPLOYEE\
WHERE\
LAST_UPDATE_DATE>TO_DATE (:Modified_Date,'YYYYMMDDHH24MISS')\
ORDER BY\
LAST_UPDATE_DATE

[SYNC-PARAMS]
CHANGEKEYATTRS: Modified_Date
```

Note that the entire `SELECT` statement is put as a value in the `SELECT` parameter in the section represented by the tag `DBQUERY`. Because it is a lengthy value, the value continuation character is put as the last character in every line until the `SELECT` statement ends.

Also note the `WHERE` condition that is present in the `SELECT` statement. The WHERE condition picks up changes based on the `Modified_Date`. To copy modified user records to the Oracle back-end directory, update the WHERE clause to pick up the records. In this example, the `Modified_Date` is the key for incremental synchronization. Because it is a date, it must be presented in string format.

The `CHANGEKEYATTRS` parameter value is the name of the columns to be used while performing incremental synchronization. The values of these columns are always stored in the `orclodipcondirlastappliedchgnum` attribute of the profile. Every time the `SELECT` statement is executed, the current values of this attribute are put into the SQL statement accordingly. This ensures that the data is always retrieved incrementally.

If there are multiple column names in the `CHANGEKEYATTRS`—for example, `column1:column2`—then the value in the `orclodipcondirlastappliedchgnum` attribute of the profile is stored as `value1~value2` and so on, with `value1` corresponding to `column1` and `value2` to `column2`.

Column names are retrieved into Oracle Directory Integration Platform as attribute value pairs and subsequently mapped into LDAP attribute values according to set mapping rules. For this reason, all columns names retrieved in the `SELECT` statement must be simple names rather than expressions. For example, you can have the expression `REPLACE(EMAIL),'@ACME.COM',''),` but it retrieves the expression value as `UID`.

When the profile is created, the `orclodipcondirlastappliedchgnum` attribute must be set to some value. All changes after this date—that is, rows in the table with `LAST_UPDATE_DATE` greater than this value—are retrieved. For example, if the `orclodipcondirlastappliedchgnum` attribute is set to `20000101000000`, then all employee changes since January 1, 2000 are retrieved.

Because of the `ORDER BY` clause, all the database rows returned are in the order of `LAST_UPDATE_DATE`—that is, the changes retrieved and applied to the directory are in chronological order. Once the last change is retrieved and applied:

1. The `orclodipcondirlastappliedchgnum` attribute value is set to the `Modified_Date` from the last row retrieved.
2. The profile is updated.

Whenever the Oracle Directory Integration Platform executes the profile again, it uses the previously stored value.

Understanding the dbexport.cfg.master Configuration File

The `dbexport.cfg.master` file describes the structure of the Oracle database. It provides the Oracle Directory Integration Platform with the following information:

- **Primary Table**. The database table that all of the other tables in the profile are connected to. Every `dbexport.cfg.master` file should have one primary table.
- **Primary Keys**. Specifies the primary key(s) for the table(s) to which you are syncing. Primary keys should be defined for all table names that are involved in this configuration.

  If a primary key consists of multiple columns, list each column name separated by a comma. For example: `id,name,dob`.
- **Table Relations**. Defines the relationships between the primary table and all of the other tables involved in the profile by specifying the connecting attribute(s). In the sample file, `id` is the connecting attribute.

  If needed, use a comma separated list to specify multiple attributes, for example: `id,name`.

The `dbexport.cfg.master` file looks like this:

```
[INTERFACEDETAILS]
Writer: oracle.ldap.odip.gsi.DatabaseWriter
CheckAllEntries: null
SkipErrorToSyncNextChange: false
UpdateSearchCount: 100
SearchDeltaSize: 500

[SYNC-PARAMS]
PRIMARY_TABLE: table1

[PRIMARY-KEYS]
table1:id
table2:id

[TABLE-RELATIONS]
table1^table2:id
```

Note: The `SkipErrorToSyncNextChange` parameter determines how the Oracle directory integration and provisioning server handles an error when processing a change during synchronization. By default, the `SkipErrorToSyncNextChange` parameter is assigned a value of false, which means that the Oracle directory integration and provisioning server will continue processing a change until the error is resolved. If you assign a value of true to the `SkipErrorToSyncNextChange` parameter, the Oracle directory integration and provisioning server will skip any changes that cause an error. Any failures are recorded in the *$ORACLE_HOME*`/ldap/odi/log/profilename.aud` audit log file. If you do assign a value of true to the `SkipErrorToSyncNextChange` parameter, be sure to periodically review the audit log for failures.

9.2 Preparing the Mapping File

The DBReader.map.master Configuration File

Follow the instructions in "Mapping rules and formats" on page 5-2 for information about configuring mapping rules.

The dbexport.map.master Configuration File

Review the instructions in "Mapping rules and formats" on page 5-2. Then open the sample file `dbexport.map.master` in the $*ORACLE_HOME*`/ldap/odi/conf` directory, and edit it to your specifications.

The mapping rule format for the `dbexport.map.master` configuration file specifies a destination table name and a destination column name instead of a destination objectclass and destination attribute.

For example, here is a sample map rule:

```
uid:1::inetorgperson:id::table2:
```

In this example, `uid` is the source attribute and `inetorgperson` is the source objectclass. Next, `id` is the destination column and `table2` is the destination table name. You must include in the map file whichever column is designated as the primary key, and you must also specify that it is a required attribute. In the example mapping rule, the `:1` following the `uid` source attribute indicates that `uid` is a required attribute.

9.3 Preparing the Directory Integration Profile

You can create a directory integration profile by using the Oracle Enterprise Manager Fusion Middleware Control user interface. For instructions, see Section 7.1, "Managing Synchronization Profiles Using Fusion Middleware Control."

When you use Oracle Enterprise Manager Fusion Middleware Control, you must upload the additional configuration information file and the mapping file by using the `update` operation of the `manageSyncProfiles` command. For help using the `manageSyncProfiles` command, see Section 7.2, "Managing Synchronization Profiles Using manageSyncProfiles."

To configure the directory integration profile, follow the general instructions in "Registering Connectors in Oracle Directory Integration Platform" on page 6-1, but with these specific instructions in mind:

- Do not set a value for the agent execution command (`orclodipAgentExeCommand`) attribute.
- Set the interface type (`orclodipDataInterfaceType`) attribute to `DB`.

9.4 Example: Synchronizing a Relational Database Table to the Back-end Directory

This section demonstrates how to synchronize a relational database table with the Oracle back-end directory.

Note: Directory Integration Platform database profiles do not support delete operations. You can, however, create a separate trigger outside of DIP that will directly delete entries in the Oracle back-end directory whenever a row is deleted in the database. Information about how to create such a trigger is outside of the scope of this documentation.

This section contains these topics:

- Configuring the Additional Configuration Information File
- Configuring the Mapping File
- Configuring the Directory Integration Profile
- Uploading the Additional Configuration Information and Mapping Files
- Synchronization Process
- Observations About the Example

In this example, the following relational database table containing employee data is synchronized with the Oracle back-end directory:

EMPNO	ENAME	LAST_UPDATE_DATE	EMAIL	TELEPHONE
98357	JOHN DOE	2-JAN-2000	JOHN.DOE@ACME.COM	435-324-3455
98360	ROGER BECK	3-JUL-2001	ROGER.BECK@ACME.COM	435-324-3600
98365	JIMMY WONG	4-MAR-2001	JIMMY.WONG@ACME.COM	435-324-2390
98370	GEORGE TWINSLEY	6-FEB-2002	GEORGE.TWINSLEY@ACME.COM	435-324-9232

You can find a sample profile (`DBReader.properties`), configuration, and mapping files for this example in the $*ORACLE_HOME*`/ldap/odi/conf` directory.

In this example:

- The name of the table is `Employee`.
- The Profile Name is `TESTDBIMPORT`.
- The employee number (`EMPNO`) is used to join a database record with a directory entry. It is specified in the OID Matching Filter (`orclOdipOIDMatchingFilter`) attribute described in the attributes reference chapter of the *Oracle Identity Management User Reference*.
- This table is present in the `testsync/testsyncpwd` schema in a database. The database is located on the host `machine.acme.com`, the database listener port is `1526`, and the SID is `iasdb`. The database URL is `machine.acme.com:1526:iasdb`.
- Appropriate read/write permissions were given explicitly to this profile, namely:
```
orclodipagentname=testdbimport,
cn=subscriber profile,
cn=changelog subscriber,
cn=oracle internet directory
```
- The profile is created in configuration set 1.

9.4.1 Configuring the Additional Configuration Information File

This example uses the same Additional Configuration Information file described earlier in "Preparing the Additional Configuration Information File" on page 9-2.

9.4.2 Configuring the Mapping File

The mapping file for this example contains the following:

```
DomainRules
NONLDAP:dc=testdbsync,dc=com:uid=%,dc=testdbsync,dc=com
AttributeRules
ename: : : :cn: :person
ename : : : :sn: :person
uid : : : :uid: :inetOrgperson:
EMail: : : :mail: :inetOrgperson
Telephone: : : :telephonenumber: :inetOrgperson
empnum: : : :employeenumber: :inetOrgperson
```

This mapping file specifies the following:

- Directory entries are created as `uid=%,dc=testdbsync,dc=com`. The percent sign (`%`) is a placeholder for the actual value of `uid`. The `uid` must be present in the mapping rules so that it has a value after the mapping. Otherwise, the DN construction fails.
- Both the `cn` and `sn` attributes need to have the same value as `ename`.
- The `uid` element must have the value of the `EMail` prefix, which is the element of the e-mail address prior to the at sign (@) character.
- The `empnum` attribute becomes `employeenumber` in the directory entry.
- The `telephone` attributes becomes `telephone number` in the directory entry.

9.4.3 Configuring the Directory Integration Profile

The directory integration profile for this example contains the attribute values as described in Table 9–1 on page 9-7. A sample integration profile with these values populated and the corresponding mapping and configuration files are available in $*ORACLE_HOME*/ldap/odi/conf directory. You can create the profile by using Oracle Enterprise Manager Fusion Middleware Control or following the instructions described in "Creating Synchronization Profiles" on page 7-1.

Table 9–1 Directory Integration Profile for TESTDBIMPORT

Attribute	Value
Profile Name (`odip.profile.name`)	`TESTDBIMPORT`
Synchronization Mode (`odip.profile.syncmode`)	`IMPORT`
Profile Status (`odip.profile.status`)	`ENABLE`
Agent Execution Command (`odip.profile.agentexecommand`)	null
Advanced Configuration Information (`odip.profile.configfile`)	Maintains configuration details which are not individually maintained in LDAP attributes.
Connected Directory Account (`odip.profile.condiraccount`)	`testdbsync`

Table 9–1 (Cont.) Directory Integration Profile for TESTDBIMPORT

Attribute	Value
Connected Directory Account Password (odip.profile.condirpassword)	testdbsyncpwd
Connected Directory URL (odip.profile.condirurl)	machine.acme.com:1526:iasdb
Interface Type (odip.profile.interface)	DB
Mapping File (odip.profile.mapfile)	Attribute for storing mapping rules.
OID Matching Filter (odip.profile.oidfilter)	employeenumber This means that employeenumber is used to search the directory while looking for a match. If a match is found, then the directory entry is modified. Otherwise, a new entry is created. This is necessary to ensure that the orclOdipOIDMatchingFilter attribute is unique in the database also. Once a database row is retrieved, the Oracle Directory Integration Platform searches the directory for that employeenumber in the domain dc=testdbsync,dc=com according to the domain rules. If it gets a match, it updates that entry with the latest values of the columns in the row retrieved. If it does not get a match, it creates a new entry in the directory with all the attributes from the column values.
Last Applied Change Number (odip.profile.lastchgnum)	20000101000000 This means that the first time the profile executes, it retrieves and synchronizes all four rows. Subsequently, it retrieves rows only when the LAST_UPDATE_DATE column in the table is updated to the time last modified.

9.4.4 Uploading the Additional Configuration Information and Mapping Files

Use the update operation of the manageSyncProfiles command to update the additional configuration information and mapping files, as follows:

```
manageSyncProfiles update -h HOST -p PORT -D WLS_USER -pf PROFILE_NAME -file FILE_
NAME
```

9.4.5 Synchronization Process

In this example, the sequence of steps in the synchronization process is:

1. The Oracle Directory Integration Platform starts a new profile thread for the TESTDBIMPORT profile every time the value specified in the scheduling interval (odip.profile.schedinterval) attribute expires.
2. The profile thread reads the additional configuration information to get the SQL to execute, and then runs the SQL.
3. For every row retrieved from the database, the mapping rules are applied to the record, and LDAP attributes are created.

4. Depending on the OID Matching Filter (`odip.profile.oidfilter`) attribute, the Oracle Directory Integration Platform determines whether a matching entry exists in the Oracle back-end directory. If it exists, then it is updated. If not, then a new entry is created. After the directory operation, the last applied change number (`odip.profile.lastchgnum`) attribute is updated.

> **Note:** The OID Matching Filter (`odip.profile.oidfilter`) attribute supports Oracle Unified Directory and Oracle Directory Server Enterprise Edition, as well as Oracle Internet Directory.

9.4.6 Observations About the Example

When a row is retrieved from the database, it is in the following form:

```
EmpNum: 98357
EName: JOHN DOE
UID: JOHN.DOE
EMAIL: JOHN.DOE@ACME.COM
TELEPHONE: 435-324-3455
Modified_Date: 20000102000000
```

After the mapping is performed on this record, the output is in the following form:

```
dn: uid=john.doe,dc=testdbsync,dc=com
uid: JOHN.DOE
cn: JOHN DOE
sn: JOHN DOE
mail: JOHN.DOE@ACME.COM
employeenumber: 98357
telephonenumber: 435-324-3455
objectclass: person
objectclass: inetorgperson
```

A subtree search is made in the directory with the filter `employeenumber=98357` under the domain `dc=testdbsync,dc=com`. If the search yields an existing entry, then that entry is updated. Otherwise, a new entry is created. Because the OID Matching Filter (`odip.profile.oidfilter`) attribute is set to `employeenumber`, every database record retrieved must have that column. In this case, it is `EmpNum` as it maps to `employeenumber`.

Any other attributes in the mapping file that are not in the data retrieved by SQL are ignored—for example, the `birthday` attribute.

After the profile thread processes all the change records from SQL, it updates the directory with correct values for these attributes:

- Last Applied Change Number (`odip.profile.lastchgnum`)
- Last Execution Time (`orclOdipLastExecutionTime`)
- Last Successful Execution Time (`orclOdipLastSuccessfulExecutionTime`)

10

Synchronizing with Oracle Human Resources

If you use Oracle Human Resources as the primary repository for employee data in your enterprise, then you may need to synchronize data between it and the Oracle back-end directory. The Oracle Human Resources connector enables you to do this.

This chapter introduces the Oracle Human Resources connector, and explains how to deploy it. It contains these topics:

- Introduction to Synchronization with Oracle Human Resources
- Data You Can Import from Oracle Human Resources
- Managing Synchronization Between Oracle Human Resources and the Oracle Back-end Directory
- The Synchronization Process
- Bootstrapping the Oracle Back-end Directory from Oracle Human Resources

---

Note: If you are synchronizing with an Oracle Human Resources environment that involves the Oracle E-Business Suite, Oracle recommends using the Oracle E-Business Suite integration solution that Oracle Directory Integration Platform supports, which is described in Chapter 15, "Integration of Provisioning Data with Oracle E-Business Suite".

See Also: The Release Notes for your Oracle back-end directory to find out which release of Oracle Human Resources can be synchronized with your Oracle back-end directory.

10.1 Introduction to Synchronization with Oracle Human Resources

The Oracle Human Resources connector enables you to import a subset of employee data from Oracle Human Resources into the Oracle back-end directory. It includes both a prepackaged integration profile and an Oracle Human Resources agent that handles communication with the Oracle back-end directory. You can customize the prepackaged integration profile to meet your deployment needs using Oracle Enterprise Manager Fusion Middleware Control.

You can schedule the Oracle Human Resources connector to run at any time, configuring it to extract incremental changes from the Oracle Human Resources system. You can also set and modify mapping between column names in Oracle Human Resources and attributes in the Oracle back-end directory.

10.2 Data You Can Import from Oracle Human Resources

Table 10–1 lists the tables in the Oracle Human Resources schema. If you choose, you can import most of these attributes into Oracle Internet Directory.

Table 10–1 Tables in Oracle Human Resources Schema

Table Name	Alias Used in the Connector Config Info Field
PER_PEOPLE_F	PER
PER_ADDRESSES	PA
PER_PERIODS_OF_ SERVICE	PPS
PER_PERSON_TYPES	PPT

All of these tables are visible if the login to the Oracle Human Resources database is done with the apps account.

Because attributes can be added or deleted at run time from the configuration file, the Oracle Human Resources connector dynamically creates a SQL statement that selects and retrieves only the required attributes.

Table 10–2 shows some of the fields in the Oracle Human Resources user interface. These fields appear when you add or modify employee data.

Table 10–2 Fields in the Oracle Human Resources User Interface

ATTRIBUTE NAME	DESCRIPTION	FORM/CANVAS/FIELD_NAME
LAST_NAME	Last name of the person	People/Name/Last
FIRST_NAME	First name of the person	People/Name/First
TITLE	Title of the person	People/Name/Title
SUFFIX	Suffix—for example, Jr, Sr, Ph.D.	People/Name/Suffix
MIDDLE_NAME	Middle name	People/Name/Middle
SEX	Sex	Gender List box
START_DATE	Hiring date	People/Hire Date
DATE_OF_BIRTH	Date of birth	People/Personal Information/Birth Date
MARITAL_STATUS	Marital status	People/Personal Information/Status
NATIONAL_ INDENTIFIER	Social security number for US residents	People/Identification/Social Security
EMPLOYEE_NUMBER	Employee number	People/Identification/Employee
REGISTERD_ DISABLED_ FLAG	Indicator that the employee has a disability	People/Personal Information/Has Disability
EMAIL_ADDRESS	Electronic mail address	People/Personal Information/EMail
OFFICE_NUMBER	Office location	People/Office Location Info/Office
MAILSTOP	Mail delivery stop	People/Office Location Info/Mail Stop
INTERNAL_ LOCATION	Location	People/Office Location Info/Location
ADDRESS_LINE1	Address line 1	Personal Address Information/Address line 1
ADDRESS_LINE2	Address line 2	Personal Address Information/Address line 2

Table 10–2 (Cont.) Fields in the Oracle Human Resources User Interface

ATTRIBUTE NAME	DESCRIPTION	FORM/CANVAS/FIELD_NAME
ADDRESS_LINE3	Address line 3	Personal Address Information/Address line 3
TOWN_OR_CITY	Town or city	Personal Address Information/City
REGION_1	First region	Personal Address Information/County
REGION_2	Second region	Personal Address Information/State
POSTAL_CODE	Postal code	Personal Address Information/Zip Code
COUNTRY	Country	Personal Address Information/Country
TELEPHONE_NUMBER_1	First telephone number	Personal Address Information/Telephone
TELEPHONE_NUMBER_2	Second telephone number	Personal Address Information/Telephone2

10.3 Managing Synchronization Between Oracle Human Resources and the Oracle Back-end Directory

This section contains these topics:

- Task 1: Configure a Directory Integration Profile for the Oracle Human Resources Connector
- Task 2: Configure the List of Attributes to be Synchronized with the Oracle Back-end Directory
- Task 3: Configure Mapping Rules for the Oracle Human Resources Connector
- Task 4: Prepare to Synchronize from Oracle Human Resources to the Oracle Back-end Directory

10.3.1 Task 1: Configure a Directory Integration Profile for the Oracle Human Resources Connector

To configure the prepackaged integration profile that is installed with the Oracle Human Resources connector, refer to Chapter 7, "Managing Directory Synchronization Profiles".

For some of the parameters in the prepackaged integration profile, you must specify values specific to integration with the Human Resources Connector. The parameters specific to the Human Resources Connector are listed in Table 10–3 on page 10-4.

Table 10–3 Attributes Specific to Oracle Human Resources Connector Integration Profile

Attribute	Description
Profile Name (`odip.profile.name`)	Unique name by which the connector is identified in the system, used as an RDN component of the DN that identifies the integration profile. The name can contain only alpha-numeric characters. This attribute is mandatory and not modifiable. The default name is `OracleHRAgent`.
Synchronization Mode (`odip.profile.syncmode`)	The direction of synchronization between the Oracle back-end directory and a connected directory. ▪ `IMPORT` indicates importing changes from a connected directory to the Oracle back-end directory. ▪ `EXPORT` indicates exporting changes from the Oracle back-end directory to a connected directory. The default is `IMPORT`. This attribute is mandatory and modifiable. **Note:** In Oracle Internet Directory 11*g* Release 1 (11.1.1), only import operations for Oracle Human Resources are supported.
Execution Information	

Agent Execution Command (`odip.profile.agentexecommand`)	Connector executable name and argument list used by the directory integration server to execute the connector.
	This attribute is mandatory and modifiable.
	The default is:
	`odihragent OracleHRAgent connect=hrdb \`
	`login=%odip.profile.condiraccount \`
	`pass=%odip.profile.condirpassword \`
	`date=%orclODIPLastSuccessfulExecutionTime \`
	You must set the value in the argument `connect=hrdb` to the connect string of the Oracle Human Resources system database.

Connected Directory Account (`odip.profile.condiraccount`)	Valid user account in the connected directory to be used by the connector for synchronization. For the Human Resources Agent, it is a valid user identifier in the Oracle Human Resources database. **See Also:** Chapter 10, "Synchronizing with Oracle Human Resources" for typical usage of passing it in the command-line
Advanced Configuration Information (`odip.profile.configfile`)	Any configuration information that you want the connector to store in the Oracle back-end directory. It is passed by the directory integration server to the connector at time of connector invocation. The information is stored as an attribute and the directory integration server does not have any knowledge of its content. The value stored in this attribute represents (for Oracle Human Resources connector) all attributes that need to be synchronized from Oracle Human Resources. **See Also:** "Task 2: Configure the List of Attributes to be Synchronized with the Oracle Back-end Directory" on page 10-5 This attribute is mandatory for the Oracle Human Resources connector.
Connected Directory URL (`odip.profile.condirurl`)	The host and port details of the connected directory. They must be entered in this format: *`host`*:*`port`*:*`sid`*.
Interface Type (`odip.profile.interface`)	The interface used for data transfer. Because it is in the form of a tagged file, it is set to `TAGGED`. **Note:** You should not modify this attribute for Oracle Human Resources profile.

Table 10–3 (Cont.) Attributes Specific to Oracle Human Resources Connector Integration Profile

Attribute	Description
Mapping Information	
Mapping Rules (`odip.profile.mapfile`)	Attribute for storing the mapping rules. Store the mapping rules in a file by using Oracle Enterprise Manager Fusion Middleware Control. See Chapter 7, "Managing Directory Synchronization Profiles" for information on using Oracle Enterprise Manager Fusion Middleware Control. This attribute is mandatory for Oracle Human Resources and is modifiable. **See Also:** ■ "Mapping rules and formats" on page 5-2 ■ "Configuring Mapping Rules" on page 6-3
Connected Directory Matching Filter (`odip.profile.condirfilter`)	This is not used in Oracle Human Resources connectivity.
OID Matching Filter (`odip.profile.oidfilter`)	This attribute names an LDAP filter that is used to search for a target entry in the Oracle back-end directory. The Oracle Directory Integration Platform uses this filter to find out what kind of LDAP operation it needs to do to synchronize. It is of the form `employeenumber=%`. It is optional and modifiable.
Status Information	
OID Last Applied Change Number (`odip.profile.lastchgnum`)	This attribute, standard for all export profiles, does not apply to Oracle Human Resources synchronization.

10.3.2 Task 2: Configure the List of Attributes to be Synchronized with the Oracle Back-end Directory

The default Oracle Human Resources profile provides a default list of attributes to be synchronized from Oracle Human Resources to the Oracle back-end directory. You can customize this list, adding attributes to it or removing attributes from it.

The default attribute list is stored in the `orclodipAgentConfigInfo` attribute as part of the integration profile.The configuration information is also available in the file `oraclehragent.cfg.master` that is located under the $*ORACLE_HOME*`/ldap/odi/conf` directory.

> **Note:** Do not modify the `oraclehragent.cfg.master` file; it serves as a backup.

Table 10–4 describes columns in the default list of Oracle Human Resources attributes.

Table 10–4 Oracle Human Resources Attributes Synchronized with the Oracle Back-end Directory by Default

Column	Description
ATTRNAME	The output tag generated in the output data file.
COLUMN_NAME	Database column name from where to obtain this value.
TABLE_NAME	Database table name from where to obtain this value.

Table 10–4 (Cont.) Oracle Human Resources Attributes Synchronized with the Oracle Back-end Directory by Default

Column	Description
FORMAT	The column data type of this attribute (ASCII, NUMBER, DATE).
MAP	Indicator of whether to extract this attribute from Oracle Human Resources. A value of `Y` indicates that it will be extracted, and a value of `N` indicates that it will not be.

The `oraclehragent.cfg.master` file contains the following:

```
ATTRNAME:COLUMN_NAME:TABLE_NAME:FORMAT:MAP
PersonId:person_id:PER:NUMBER:Y
PersonType:person_type_id:PER:NUMBER:Y
PersonTypeName:system_person_type:PPT:ASCII:Y
LastName:last_name:PER:ASCII:Y
StartDate:start_date:PER:DATE:Y
BirthDate:date_of_birth:PER:DATE:Y
EMail:email_address:PER:ASCII:Y
EmployeeNumber:employee_number:PER:NUMBER:Y
FirstName:first_name:PER:ASCII:Y
FullName:full_name:PER:ASCII:Y
knownas:known_as:PER:ASCII:Y
MaritalStatus:marital_status:PER:ASCII:Y
middleName:middle_names:PER:ASCII:Y
country:country:PA:ASCII:Y
socialsecurity:national_identifier:PER:ASCII:Y
Sex:sex:PER:ASCII:Y
Title:title:PER:ASCII:Y
suffix:suffix:PER:ASCII:Y
street1:address_line1:PA:ASCII:Y
zip:postal_code:PA:ASCII:Y
Address1:address_line1:PA:ASCII:Y
Address2:address_line2:PA:ASCII:Y
Address3:address_line3:PA:ASCII:Y
TelephoneNumber1:telephone_number_1:PA:ASCII:Y
TelephoneNumber2:telephone_number_2:PA:ASCII:Y
TelephoneNumber3:telephone_number_3:PA:ASCII:Y
town_or_city:town_or_city:PA:ASCII:Y
state:region_2:PA:ASCII:Y
Start_date:effective_start_date:PER:DATE:Y
End_date:effective_end_date:PER:DATE:Y
per_updateTime:last_update_date:PER:DATE:Y
pa_updateTime:last_update_date:PA:DATE:Y
```

10.3.2.1 Modifying Additional Oracle Human Resources Attributes for Synchronization

To include additional Oracle Human Resources attributes for synchronization, follow these steps:

1. Copy the `oraclehragent.cfg.master` file and name it anything other than *Agent_Name*`.cfg`. This is because the Oracle Directory Integration Platform generates a configuration file with that name, using it to pass the configuration information to the Oracle Human Resources agent at run time.
2. Include an additional Oracle Human Resources attribute for synchronization by adding a record to this file.

To do this, you need this information:

- Table name in the database from which the attribute value is to be extracted. These tables are listed in Table 10–1 on page 10-2. The file uses abbreviated names for the four tables used in the synchronization.
- Column name in the table.
- Column data type. Valid values are `ASCII`, `NUMBER`, and `DATE`.

You also need to assign an attribute name to the column name. This acts as the output tag that is used to identify this attribute in the output file. This tag is used in the mapping rules to establish a rule between the Oracle Human Resources attribute and the Oracle back-end directory attribute.

You must also ensure that the `map` column—that is, the last column in the record—is set to the value `Y`.

Note: If you add a new attribute in the attribute list, then you must define a corresponding rule in the `orclodipAttributeMappingRules` attribute. Otherwise the Oracle Human Resources attribute is not synchronized with the Oracle back-end directory, even if it is being extracted by the Oracle Human Resources connector.

10.3.2.2 Excluding Oracle Human Resources Attributes from Synchronization

To exclude an Oracle Human Resources attribute that is currently being synchronized with Oracle Internet Directory, do the following:

1. Copy the `oraclehragent.cfg.master` file and name it anything other than *Agent_Name*`.cfg`. This is because the directory integration server generates a configuration file with that name, using it to pass the configuration information to the Oracle Human Resources connector at run time.
2. Do one of the following:
   - Comment out the corresponding record in the attribute list by putting a number sign (#) in front of it.
   - Set the value of the column `map` to `N`.

10.3.2.3 Configuring a SQL SELECT Statement in the Configuration File to Support Complex Selection Criteria

If the previous supporting attribute configuration is not sufficient to extract data from the Oracle Human Resources database, then the Oracle Human Resources agent can also execute a preconfigured SQL `SELECT` statement in the configuration file. There is a tag to indicate this in the configuration file, namely, a `[SELECT]` in the configuration file.

The following example shows a sample select statement to retrieve some information from the Oracle Human Resources database. Note that only the SQL statement should follow the `[SELECT]` tag. The `BINDVAR` bind variable needs to be there to retrieve incremental changes. The substitutes passes this value (the time stamp) to the Oracle Human Resources connector.

All the column expressions retrieved in the `SELECT` statement must have column names—for example, `REPLACE(ppx.email_address),'@ORACLE.COM','')` is retrieved as `EMAILADDRESS`. The Oracle Human Resources connector writes out

EMAILADDRESS as the attribute name in the output file with its value as the result of the expression REPLACE(ppx.email_address),'@ORACLE.COM''').

The following is an example of a a SELECT statement in a configuration file.

```
[SELECT]

SELECT
     REPLACE(ppx.email_address,'@ORACLE.COM','') EMAILADDRESS ,
     UPPER(ppx.attribute26) GUID,
     UPPER(ppx.last_name) LASTNAME,
     UPPER(ppx.first_name) FIRSTNAME,
     UPPER(ppx.middle_names) MIDDLENAME,
     UPPER(ppx.known_as) NICKNAME,
     UPPER(SUBSTR(ppx.date_of_birth,1,6)) BIRTHDAY,
     UPPER(ppx.employee_number) EMPLOYEEID,
     UPPER(ppos.date_start)  HIREDATE
FROM
   hr_organization_units hou,
   per_people_x ppx,
   per_people_x mppx,
   per_periods_of_service ppos
  WHERE
      pax.supervisor_id  = mppx.person_id(+)
  AND pax.organization_id = hou.organization_id(+)
  AND ppx.person_id = ppos.person_id
  AND ppx.person_id = pax.person_id
  AND ppos.actual_termination_date IS NULL
  AND UPPER(ppx.current_employee_flag) = 'Y'
  AND ppx.last_update_date >= (:BINDVAR,'YYYYMMDDHH24MISS')
```

10.3.3 Task 3: Configure Mapping Rules for the Oracle Human Resources Connector

Attribute mapping rules govern how the Oracle Directory Integration Platform converts attributes between Oracle Human Resources and Oracle Internet Directory. You can customize the mapping rules you want the Oracle Directory Integration Platform to use.

The Oracle Human Resources agent profile has a default mapping file with a set of mapping rules in the attribute orclodipAttributeMappingRules. This information is also stored in the file named oraclehragent.map.master located under the $*ORACLE_HOME*/ldap/odi/conf directory.

Note: Do not modify the oraclehragent.map.master file. It serves as a backup.

See Also: "Mapping rules and formats" on page 5-2 for the contents of the oraclehragent.map.master file and a description of the format of the mapping rules records

10.3.4 Task 4: Prepare to Synchronize from Oracle Human Resources to the Oracle Back-end Directory

This section explains how to set up synchronization from Oracle Human Resources to the Oracle back-end directory.

10.3.4.1 Preparing for Synchronization

To prepare for synchronization between Oracle Human Resources and Oracle Internet Directory, follow these steps:

1. Ensure that the Oracle Human Resources connector and the directory integration server are installed on the host from which you want to run the Oracle Human Resources connector.
2. Ensure that you have the information for accessing the Oracle Human Resources system, including:
   - Connect string to the Oracle Human Resources system database
   - Access account
   - Password
3. Configure an integration profile for the Oracle Human Resources connector, as described in "Task 1: Configure a Directory Integration Profile for the Oracle Human Resources Connector" on page 10-3. Ensure that all values in the integration profile are properly set, including:
   - Oracle Human Resources attribute list
   - Oracle Human Resources attribute mapping rules
   - Scheduling interval
4. Once everything is properly set, set the Profile Status attribute to `ENABLE`. This indicates that the Oracle Human Resources connector is ready to run.
5. Start the Oracle directory server and the Oracle Human Resources system if they are not already running on the respective hosts.
6. When everything is ready, start the directory integration server if it is not already running on this host.

10.4 The Synchronization Process

Once the Oracle Human Resources system, Oracle Internet Directory, and the Oracle Directory Integration Platform are running, and the Oracle Human Resources connector is enabled, the Oracle Directory Integration Platform automatically starts synchronizing changes from the Oracle Human Resources system into Oracle Internet Directory. It follows this process:

1. Depending on the value specified in the Last Execution Time (`orclodipLastExecutionTime`) and the Scheduling Interval (`orclodipschedulinginterval`), the Oracle Directory Integration Platform invokes the Oracle Human Resources connector.
2. The Human Resources agent extracts:
   - All the changes from the Oracle Human Resources System based on the time specified in the `orclodipLastSuccessfulExecutionTime` attribute in the integration profile
   - Only the attributes specified in the `orclodipAgentConfigInfo` attribute in the profile

   It then writes the changes into the Oracle Human Resources import file, namely $*ORACLE_HOME*`/ldap/odi/import/HR_Agent_Name.dat`.

3. After the agent completes execution, it creates a data file that looks similar to the following:

```
FirstName: John
LastName: Liu
EmployeeNumber: 12345
Title: Mr.
Sex: M
MaritalStatus: Married
TelephoneNumber: 123-456-7891
Mail: Jliu@my_company.com
Address: 100 Jones Parkway
City: MyTown
```

4. The Oracle Directory Integration Platform imports the changes to the Oracle back-end directory by doing the following:
   - Reading each change record from the import file.
   - Converting each change record into an LDAP change entry based on the rules specified in the Mapping Rules (`orclodipAttributeMappingRules`) in the integration profile.
5. After importing all the changes to the Oracle back-end directory, Oracle Human Resources connector moves the import file to the archive directory, `$`*`ORACLE_HOME`*`/ldap/odi/import/archive`. The status attributes Last Execution Time (`orclodipLastExecutionTime`) and Last Successful Execution Time (`orclodipLastSuccessfulExecutionTime`) are updated to the current time.

   If the import operation fails, only the Last Execution Time (`orclodipLastExecutionTime`) attribute is updated, and the connector attempts to extract the changes from Human Resources system based on the Last Successful Execution Time (`orclodipLastSuccessfulExecutionTime`) attribute. The reason for failure is logged in the trace file in $*ORACLE_HOME*/ldap/odi/HR_Agent_Name.trc file.

10.5 Bootstrapping the Oracle Back-end Directory from Oracle Human Resources

There are two ways to bootstrap Oracle Internet Directory from Oracle Human Resources:

- Use the Oracle Human Resources connector. In the integration profile, set the `orclodipLastSuccessfulExecutionTime` attribute to a time before Oracle Human Resources was installed.
- Use external tools to migrate data from Oracle Human Resources into Oracle Internet Directory.

11

Synchronizing with Third-Party Metadirectory Solutions

To enable synchronization with supported third-party metadirectory solutions, the Oracle back-end directory uses change logs. The Oracle Directory Integration Platform does not provide mapping or scheduling services for third-party metadirectory solutions.

This chapter describes how change log information is generated and how supporting solutions use that information. It tells you how to enable third-party metadirectory solutions to synchronize with the Oracle back-end directory.

This chapter contains these topics:

11.1 About Change Logs

The Oracle back-end directory records each change as an entry in the change log container. A third-party metadirectory solution retrieves changes from the change log container and applies them to the third-party directory. To retrieve these changes, the third-party metadirectory solution must subscribe to the Oracle back-end directory change logs.

Each entry in the change log has a change number. The third-party metadirectory solution keeps track of the number of the last change it applied, and it retrieves from the Oracle back-end directory only those changes with numbers greater than the last change it applied. For example, if the last change a third-party metadirectory solution retrieved was a number of 250, then subsequent changes it retrieves would be greater than 250.

Note: If a third-party metadirectory solution is not subscribed to the Oracle back-end directory change logs, and the first change it retrieves is more than one number higher than the last change it last applied, then some of the changes in the Oracle back-end directory change log have been purged. In this case, the third-party metadirectory solution must read the entire Oracle back-end directory to synchronize its copy with that in the Oracle back-end directory.

See Also: "Components Involved in Oracle Directory Synchronization" on page 5-1 for a conceptual discussion of directory integration profiles

11.2 Enabling Third-Party Metadirectory Solutions to Synchronize with the Oracle Back-end Directory

To enable third-party metadirectory solutions to retrieve changes from the Oracle back-end directory, perform the tasks described in this section.

- Task 1: Perform Initial Bootstrapping
- Task 2: Create a Change Subscription Object in the Oracle Back-end Directory for the Third-Party Metadirectory Solution

11.2.1 Task 1: Perform Initial Bootstrapping

To bootstrap a directory to synchronize data between a local directory and the Oracle back-end directory, do the following:

1. Find the number of the last change recorded in the Oracle back-end directory. This number is in the DSE root attribute, `lastChangeNumber`.

   To find the number of the last change recorded in the Oracle back-end directory, use the `ldapsearch` command. Enter the following command:

   ```
   ldapsearch -h host_name -p port_number -D binddn -q -s base \
   -b "" 'objectclass=*' lastchangenumber
   ```

   If the change log does not contain change entries because they have been purged, then the last change number retrieved is `0` (zero).

2. Use the `ldifwrite` command to export data from the Oracle back-end directory into an LDIF file.

3. Convert the LDIF file to a format suitable to the client directory, then load it into the client directory.

Note: Initial bootstrapping is not required with a new installation of the Oracle back-end directory. In this case, the current change number of the newly installed Oracle back-end directory is `0` (zero).

See Also: If your Oracle back-end directory is Oracle Internet Directory, see the `ldifwrite` section in the Oracle Internet Directory data management tools chapter of the *Oracle Identity Management User Reference.*

11.2.2 Task 2: Create a Change Subscription Object in the Oracle Back-end Directory for the Third-Party Metadirectory Solution

To enable a third-party metadirectory solution to synchronize with the Oracle back-end directory, you must create a change subscription object for it in the Oracle back-end directory. This gives the third-party metadirectory solution access to change log objects stored in the Oracle back-end directory.

11.2.2.1 About the Change Subscription Object

If Oracle Internet Directory is the Oracle back-end directory, the change subscription object is an entry located under the following container:

```
cn=Subscriber Profile,cn=ChangeLog Subscriber,cn=Oracle Internet Directory
```

If Oracle Unified Directory or Oracle Directory Server Enterprise Edition is the Oracle back-end directory, the change subscription object is an entry located under the following container:

```
cn=Subscriber Profile,cn=ChangeLog Subscriber,cn=Directory Integration Platform,
<suffix>
```

This change subscription object provides a unique credential for a third-party metadirectory solution to bind with the Oracle back-end directory and to retrieve changes from it. You associate the change subscription object with the auxiliary object class `orclChangeSubscriber`. This object class has several attributes, of which the following are mandatory:

- `userPassword`

  Password to be used by the directory when accessing the change log object in the Oracle back-end directory.

- `orclLastAppliedChangeNumber`

  Number of the change applied during the last synchronization. This attribute allows the directory to retrieve only the changes in the Oracle back-end directory it has not already applied.

11.2.2.2 Creating a Change Subscription Object

To create a change subscription object, use the `ldapadd` command. The following example uses an input file, named `add.ldif`, to create and enable a change subscription object, named `my_change_subscription_object`, under the following container:

- If Oracle Internet Directory is the Oracle back-end directory:

  ```
  cn=Subscriber Profile,cn=ChangeLog Subscriber,cn=Oracle Internet
  Directory
  ```

- If Oracle Unified Directory or Oracle Directory Server Enterprise Edition is the Oracle back-end directory:

  ```
  cn=Subscriber Profile,cn=ChangeLog Subscriber,cn=Directory Integration
  Platform, <suffix>
  ```

The `orclLastAppliedChangeNumber` attribute is the current change number in the directory before initial bootstrapping—in this example, 250.

- Edit the `add.ldif` file (in this example Oracle Internet Directory is the back-end directory):

```
dn: cn=my_change_subscription_object,cn=Subscriber Profile,
  cn=ChangeLog Subscriber,cn=Oracle Internet Directory
userpassword: my_password
orclLastAppliedChangeNumber: 250
orclSubscriberDisable: 0
objectclass: orclChangeSubscriber
objectclass: top
```

- Add the entry:

```
ldapadd -h my_host -D binddn -q -p PORT -f add.ldif
```

> **See Also:** "Disabling and Deleting Change Subscription Objects" on page 11-5 for instructions about temporarily disabling or deleting change subscription objects

11.3 Synchronization Process

This section contains these topics:

- How a Connected Directory Retrieves Changes the First Time from the Oracle Back-end Directory
- How a Connected Directory Updates the orclLastAppliedChangeNumber Attribute in the Oracle Back-end Directory

11.3.1 How a Connected Directory Retrieves Changes the First Time from the Oracle Back-end Directory

In this example, a connected directory with a change subscription object named `my_change_subscription_object` acquires changes from the Oracle back-end directory.

```
ldapsearch -h my_host -D binddn -q -p PORT -b "cn=changeLog" -s one
(&(objectclass=changeLogEntry)
(changeNumber >= orclLastAppliedChangeNumber )
( ! (modifiersname =cn=my_change_subscription_object,cn=Subscriber Profile,
     cn=ChangeLog Subscriber,cn=Oracle Internet Directory ) ) )
```

When the directory is retrieving changes for the first time, the value for `orclLastAppliedChangeNumber` is the number you set in "Task 2: Create a Change Subscription Object in the Oracle Back-end Directory for the Third-Party Metadirectory Solution" on page 11-3.

The `(!(modifiersname=`*client_bind_dn*`))` argument in the filter ensures that the Oracle back-end directory does not return changes made by the connected directory itself.

11.3.2 How a Connected Directory Updates the orclLastAppliedChangeNumber Attribute in the Oracle Back-end Directory

After retrieving changes from the Oracle back-end directory, the connected directory updates the `orclLastAppliedChangeNumber` attribute in its change subscription object in the Oracle back-end directory. This allows the Oracle back-end directory to purge changes that connected directories have already applied. It also enables the

connected directory to retrieve only the most recent changes, ignoring those it has already applied.

This example uses an input file, `mod.ldif`, in which the connected directory has a change subscription object named `my_change_subscription_object`, and the last applied change number is 121. The connected directory updates `orclLastAppliedChangeNumber` in its change subscription object in the Oracle back-end directory as follows:

1. Edit the `mod.ldif` file:

```
dn: cn=my_change_subscription_object,cn=Subscriber Profile,
    cn=ChangeLog Subscriber,cn=Oracle Internet Directory
changetype:modify
replace: orclLastAppliedChangeNumber
orclLastAppliedChangeNumber: 121
```

2. Use the `ldapmodify` command to load the edited mod.ldif file:

```
ldapmodify -h host -D binddn -q -p port -f mod.ldif
```

> **See Also:** If Oracle Internet Directory is your Oracle back-end directory, see the chapter about garbage collection in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about purging changes according to change numbers.

11.4 Disabling and Deleting Change Subscription Objects

You can temporarily disable or delete an existing change subscription object. This section contains these topics:

- Disabling a Change Subscription Object
- Deleting a Change Subscription Object

11.4.1 Disabling a Change Subscription Object

If a change subscription object already exists for a third-party metadirectory solution, but you want to disable it temporarily, then set the `orclSubscriberDisable` attribute to `1`. The following example uses an input file, `mod.ldif`, to disable a change subscription object.

- Edit the `mod.ldif` file (in this example Oracle Internet Directory is the back-end directory):

```
dn: cn=my_change_subscription_object,cn=Subscriber Profile,
    cn=ChangeLog Subscriber,cn=Oracle Internet Directory
changetype: modify
replace: orclSubscriberDisable
orclSubscriberDisable: 1
```

- Modify the entry:

```
ldapmodify -h my_ldap_host -D binddn -q -p PORT -v -f mod.ldif
```

11.4.2 Deleting a Change Subscription Object

To delete a change subscription object, use the `ldapdelete` command. Enter the following command (in this example Oracle Internet Directory is the back-end directory):

```
ldapdelete -h ldap_host -D binddn -q -p ldap_port
```

```
"cn=my_change_subscription_object,cn=Subscriber Profile,
 cn=ChangeLog Subscriber,cn=Oracle Internet Directory"
```

Part IV

Provisioning with the Oracle Directory Integration Platform

This part discusses the concepts and components involved in provisioning and the process through which an application receives changes to user or group entries or attributes that it needs to track. It contains these chapters:

12

Understanding the Oracle Directory Integration Platform for Provisioning

As of 11*g* Release 1 (11.1.1), Oracle offers two complementary provisioning products, optimized for different use cases:

- Oracle Identity Manager is an enterprise provisioning platform designed to manage complex environments with highly heterogeneous technologies that can include directories, databases, mainframes, proprietary technologies, and flat files. Oracle Identity Manager offers full-functioned workflow and policy capabilities along with a rich set of audit and compliance features.
- Oracle Directory Integration Platform, a component of the Identity Management infrastructure, is a meta-directory technology designed to perform directory synchronization as well as provisioning tasks in a directory-centric environment. Oracle Directory Integration Platform is designed to manage a more homogeneous environment consisting of directories and compatible Oracle products. Oracle Directory Integration Platform performs provisioning tasks by using data synchronization and offers a small deployment footprint when workflow and a full feature policy engine are not required.

> **Note:** You must use Oracle Internet Directory as your back-end directory server to use the Oracle Directory Integration Platform provisioning functionality.

This chapter discusses Oracle Directory Integration Platform Provisioning. It contains these sections:

See Also:

- The chapter on developing provisioning-integrated applications in *Oracle Fusion Middleware Application Developer's Guide for Oracle Identity Management*

12.1 What Is Provisioning?

Provisioning refers to the process of providing users, groups, and other objects with access to applications and other resources that may be available in an enterprise environment. A provisioning-integrated application refers to an application that has registered for provisioning events and registered a provisioning-integration profile in the Oracle Internet Directory back-end directory. At times, you may want to synchronize all user entries in an application-specific directory with those in the Oracle Internet Directory back-end directory, but provision a particular application to receive notification about only some of them. For example, the directory for Oracle Human Resources typically contains data for all employees in an enterprise, and you would probably want to synchronize all of that data with the Oracle Internet Directory back-end directory. However, you might want to provision another application, such as Oracle Email, to be notified only when members join or leave a particular group.

Before a user account can be provisioned for applications in an Oracle Identity Management deployment, it must first be created in the Oracle Internet Directory back-end directory. User accounts can be created in the Oracle back-end directory with the following tools or methods:

- Oracle Internet Directory Provisioning Console
- Directory Integration Assistant's `bulkprov` operation
- Synchronization with other Oracle and third-party directories
- Command-line LDAP tools

The Oracle Directory Integration Platform Service can be invoked for any user entries, regardless of how they were created in the Oracle Internet Directory back-end directory. However, creating a user entry in the Oracle back-end directory does not necessarily mean that the user entry will have access to all applications in the Oracle Identity Management environment. The user account must be manually provisioned by an administrator or automatically provisioned according to an application's provisioning policies. The default provisioning policy of an application can be one of the following:

- Provision all users
- Do not provision users
- Provision users after evaluating a provisioning policy

Provisioning policies are entirely dependent on the needs and requirements within each enterprise environment. For example, an organization may choose to provision all users with access to an e-mail application, but may restrict the users that are provisioned to access a human resources application.

12.2 Components of the Oracle Directory Integration Platform Service

The Oracle Directory Integration Platform Service consists of the following components:

- The Oracle Directory Integration Platform.

  See Also: Chapter 4, "Managing the Oracle Directory Integration Platform"

- A provisioning integration profile for each provisioning-integrated application in which you want to provision users. You create a provisioning-integration profile by using the `oidprovtool`.

See Also: "Managing Provisioning Profiles Using oidprovtool" on page 13-2 for information about `oidprovtool`.

12.3 Understanding Provisioning Concepts

This section explains how applications are provisioned with Oracle Directory Integration Platform Provisioning. It contains these topics:

- Synchronous Provisioning
- Asynchronous Provisioning
- Provisioning Data Flow

12.3.1 Synchronous Provisioning

A provisioning-integrated application can maintain user information in the Oracle Internet Directory back-end directory or a connected repository. Applications that maintain user information in Oracle Internet Directory can use the Data Access Java plug-in to create, modify, and delete user entries whenever the change occurs in Oracle Internet Directory.

See Also: *Oracle Fusion Middleware Application Developer's Guide for Oracle Identity Management* for more information about the Data Access Java plug-in

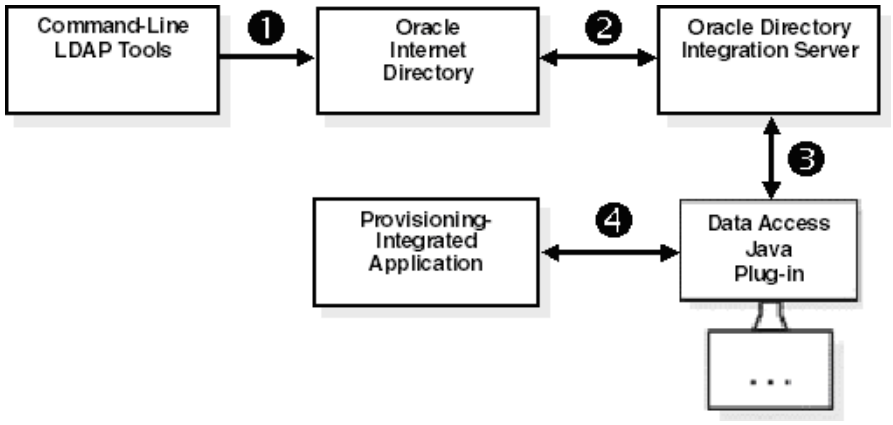
The Data Access Java plug-in can be invoked directly from Oracle Identity Management, including the command-line LDAP tools. For this reason, applications that can be provisioned with the Data Access Java plug-in are provisioned synchronously; no separate provisioning event needs to be sent to the application from the Oracle Directory Integration Platform. The Data Access Java plug-in returns either `SUCCESS` or `FAILURE` to the Oracle Directory Integration Platform. If an execution status of `SUCCESS` is returned for the Data Access Java plug-in, then a provisioning status is also returned, which is recorded in the user's provisioning status attribute in Oracle Internet Directory for the specific provisioning-integrated application. If the status of `FAILURE` is returned for new user provisioning requests, then the user's provisioning status is assigned a value of `PROVISIONING_FAILURE`. See "Provisioning Status in Oracle Internet Directory" on page 12-11 for a list of provisioning statuses.

Synchronous provisioning follows this process:

1. A new user entry is created in Oracle Internet Directory from one of the following sources:
   - The Oracle Enterprise Manager user interface
   - Bulk provisioning with the Directory Integration Assistant
   - Synchronization with connected Oracle and third-party directories
2. The Oracle Identity Management component that created the new user entry invokes the Data Access Java plug-in.
3. The Data Access Java plug-in provisions the new user account in the application.

Figure 12–1 illustrates the process of how an application is synchronously provisioned using command-line LDAP tools.

Figure 12–1 Synchronous Provisioning from Command-Line LDAP Tools



Synchronous provisioning from command-line LDAP tools follows this process:

1. A command-line LDAP tool creates a new user entry in the Oracle Internet Directory back-end directory.
2. At the next scheduled synchronization interval, the Oracle Directory Integration Platform identifies new user entries in Oracle Internet Directory that require provisioning.
3. The Oracle Directory Integration Platform invokes the Data Access Java plug-in.
4. The Data Access Java plug-in provisions the new user accounts in the application.

12.3.2 Asynchronous Provisioning

The Oracle Directory Integration Platform propagates PL/SQL events to a provisioning-integrated application, which then executes a PL/SQL plug-in to process the events. Execution of a PL/SQL plug-in occurs within the application repository and not within the address space of any Oracle Identity Management component. Because, provisioning is handled by a PL/SQL plug-in and not by any component of Oracle Identity Management, provisioning-integrated applications that implement a PL/SQL plug-in are provisioned asynchronously. The PL/SQL plug-in returns the status of `SUCCESS` or `FAILURE` to the Oracle Directory Integration Platform. If the status of `SUCCESS` is returned for the PL/SQL plug-in, then a provisioning status is also returned, which is recorded in the user's provisioning status attribute in Oracle Internet Directory for the specific provisioning-integrated application. If the status of `FAILURE` is returned for new user provisioning requests, then the user's provisioning status is assigned a value of `PROVISIONING_FAILURE`. See "Provisioning Status in Oracle Internet Directory" on page 12-11 for a list of provisioning statuses.

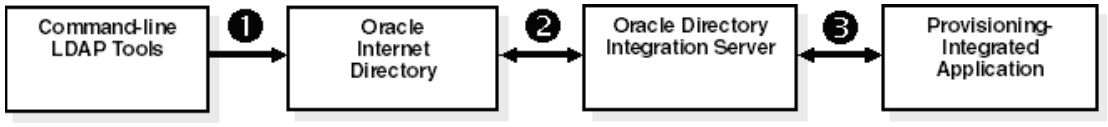
Asynchronous provisioning follows this process:

1. A new user entry and an associated entry containing application-specific user preferences are created in Oracle Internet Directory from one of the following sources:
   - Oracle Enterprise Manager user interface
   - Bulk provisioning with the provProfileBulkProv command
   - Synchronization with connected Oracle and third-party directories

2. At the next scheduled synchronization interval, the Oracle Directory Integration Platform identifies new user entries in Oracle Internet Directory that require provisioning.
3. Provisioning events are sent from the Oracle Directory Integration Platform to the PL/SQL plug-in.

Figure 12–2 illustrates the process of how an application is asynchronously provisioned using command-line LDAP tools.

Figure 12–2 Asynchronous Provisioning using Command-Line LDAP Tools



As illustrated in Figure 12–2, asynchronous provisioning using command-line LDAP tools follows this process:

1. A new user entry is created in the Oracle Internet Directory back-end directory using a command-line LDAP tool.
2. At the next scheduled synchronization interval, the Oracle Directory Integration Platform identifies new users entries in Oracle Internet Directory that require provisioning, and creates an associated entry containing application-specific user preferences.
3. Provisioning events are sent from the Oracle Directory Integration Platform to the PL/SQL plug-in.

12.3.3 Provisioning Data Flow

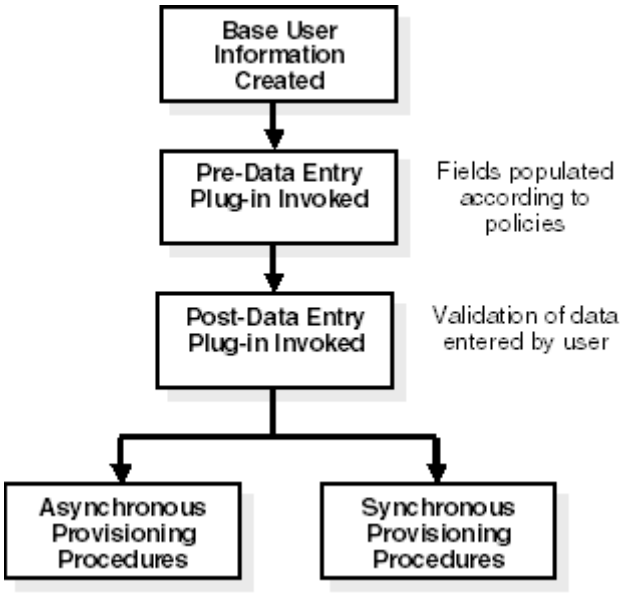
Regardless of whether it is provisioned synchronously or asynchronously, an application can invoke the Pre-Data Entry and Post-Data Entry plug-ins to enhance provisioning intelligence and implement business policies. Both plug-ins are invoked by Oracle Identity Management components such as the Oracle Internet Directory Provisioning Console and bulk provisioning with the provProfileBulkProv command.

The Pre-Data Entry plug-in populates fields according to provisioning policies. The primary purpose of this plug-in is to determine whether a user should be provisioned in an application. For example, if an organization has a a policy where only managers are provisioned for a financial application, the Pre-Data Entry plug-in can be used to identify which user entries to provision. Common user attributes are already populated when this plug-in is invoked, so it should have adequate information to make provisioning decisions.

The Post-Data Entry plug-in primarily validates data entered by users for common attributes and application-specific attributes. The validation for the plug-in must be successful for provisioning to continue.

Figure 12–3 illustrates the provisioning data flow using the Pre-Data Entry and Post-Data Entry plug-ins.

Figure 12–3 Provisioning Data Flow



This figure conceptually illustrates the provisioning data flow process. An arrow points from a Base User Information Created box at the top of the illustration to a Pre-Data Entry Plug-in Invoked box. Another arrow points from the Pre-Data Entry Plug-in Invoked box to Post-Data Entry Plug-in Invoked box. Two arrows point from the Post-Data Entry Plug-in Invoked box to an Asynchronous Provisioning Procedures box and a Synchronous Procedures box.

****

As illustrated in Figure 12–3, the provisioning data flow follows this process:

1. Base user information is created.
2. The Pre-Data Entry plug-in is invoked, which populates fields according to policies.
3. The Post-Data Entry plug-in is invoked, which validates data entered by the user.
4. Depending on the provisioning approach, either asynchronous or synchronous provisioning procedures are invoked.

12.4 Overview of Provisioning Methodologies

This section describes the procedures for provisioning users in Oracle Identity Management. It contains these topics:

- Provisioning Users that are Synchronized from an External Source
- Provisioning Users Created with Command-Line LDAP Tools
- Bulk Provisioning Using the provProfileBulkProv Tool
- On-Demand Provisioning
- Application Bootstrapping

12.4.1 Provisioning Users that are Synchronized from an External Source

When Oracle Internet Directory is used as a central repository, and enterprise user entries are synchronized from connected directories to the Oracle Internet Directory back-end directory, each user identity is automatically provisioned according to the default provisioning policy of each provisioning-integrated application.

12.4.2 Provisioning Users Created with Command-Line LDAP Tools

Any tools developed by Oracle or third-party vendors that use standard command-line LDAP syntax can create user entries in Oracle Internet Directory. As with user entries that are synchronized from external sources, any user entries created with command-line LDAP tools or any other means are provisioned according to the default provisioning policies for each provisioning-integrated application.

12.4.3 Bulk Provisioning Using the provProfileBulkProv Tool

Use the `provProfileBulkProv` utility, located in the *ORACLE_HOME*/bin directory, to perform initial migration of data from an LDIF file to Oracle Internet Directory for a provisioning profile.

Notes:

- Best security practice is to provide a password only in response to a prompt from the command.
- You must set the `WLS_HOME` and `ORACLE_HOME` environment variables before executing any of the Oracle Directory Integration Platform commands
- The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. Refer to the *Configuring SSL* chapter in *Oracle Fusion Middleware Securing Oracle WebLogic Server* for more information.

12.4.3.1 Syntax for provProfileBulkProv

provProfileBulkProv

```
provProfileBulkProv -h HOST -p PORT -D wlsuser -file LDIF_FILE -realm REALM_DN
[-ssl -keystorePath PATH_TO_KEYSTORE -keystoreType TYPE]
[-encoding INPUT_ENCODING] [-help]
```

12.4.3.2 Arguments for provProfileBulkProv

-h | -host

Oracle WebLogic Server host where Oracle Directory Integration Platform is deployed.

-p | -port

Listening port of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed.

-D | -wlsuser

Oracle WebLogic Server login ID

> **Note:** You will be prompted for the Oracle WebLogic Server login password. You cannot provide the password as a command-line argument. Best security practice is to provide a password only in response to a prompt from the command. If you must execute `provProfileBulkProv` from a script, you can redirect input from a file containing the Oracle WebLogic Server login password. Use file permissions to protect the file and delete it when it is no longer necessary.

-f | -file

LDIF file containing the data to be migrated.

-realm

The realm in which the users are to be provisioned.

-ssl

Executes the command in SSL mode.

> **Note:** The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. Refer to the *Configuring SSL* chapter in *Oracle Fusion Middleware Securing Oracle WebLogic Server* for more information.

-keystorePath

The full path to the keystore.

-keystoreType

The type of the keystore identified by `-keystorePath`. For example: `-keystorePath jks` or `-keystorePath PKCS12`

-encoding

Input file encoding.

-help

Provides command usage help.

12.4.3.3 Tasks and Examples for provProfileBulkProv

```
provProfileBulkprov -h myhost.mycompany.com -p 7005 -D login_ID \
  -f /opt/ldap/odip/users.ldif -realm cn=aaaa,ou=bbbb,dc=cccc
```

12.4.4 On-Demand Provisioning

On-demand provisioning occurs when a user accesses an application and the application has no knowledge of the user in its repository. The application determines whether to provision a user account based on its default provisioning policies. After

provisioning a user account in its repository, an application will update the provisioning status of the user entry in the Oracle Internet Directory back-end directory.

12.4.5 Application Bootstrapping

The Oracle Directory Integration Platform Service notifies newly registered applications of all existing user entries in Oracle Internet Directory and attempts to provision each existing user entry as if they were a new user in the application.

12.5 Organization of User Profiles in the Oracle Internet Directory Back-End Directory

This section discusses the organization of user profiles in the Oracle Internet Directory back-end directory. It contains these topics:

- Organization of Provisioning Entries in the Directory Information Tree
- Understanding User Provisioning Statuses

12.5.1 Organization of Provisioning Entries in the Directory Information Tree

The Oracle Directory Integration Platform Provisioning relies on user profiles in the directory information tree (DIT) that consist of attributes containing personal information and preferences for the various applications in which the user is provisioned. These user attributes for the Oracle Directory Integration Platform Service can be categorized as follows:

- Base attributes that are available for every user entry
- Application-specific attributes that are only available if a user is provisioned in an application

Base user attributes primarily belong to standard LDAP object classes such as `organizationalPerson` and `inetOrgPerson`, and consist of personal details that include first name, last name, given name, e-mail address, and telephone numbers. Base user attributes also consist of Oracle application-specific attributes that belong to the `orclUserV2` auxilliary class.

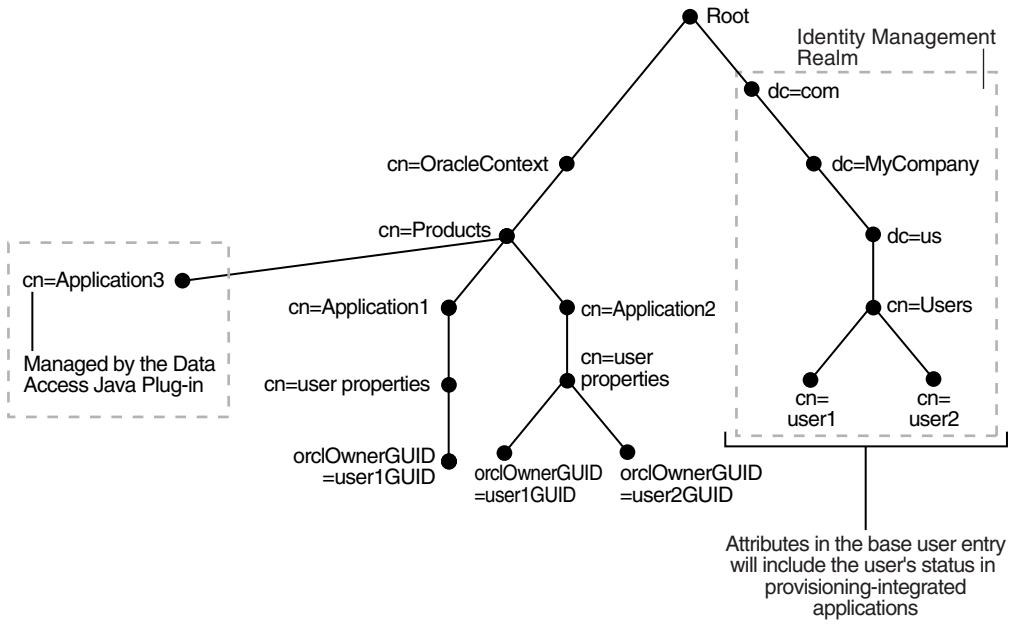
The Oracle Internet Directory back-end directory is the primary repository for both base attributes and application-specific attributes. Both types of attributes are stored in each user's profile. However, an application can cache user attributes that are updated with the provisioning event notification service.

As shown in Figure 12–4, user attributes are stored in two locations within the DIT. Base user entries, which include attributes belonging to `inetorgperson` and `orcluserv2`, are stored under `cn=users,`*`Realm DN`*. The provisioning status of each user entry is also stored in the base user entry. Application-specific attributes reside in separate entries in the application container. The LDAP schema relating to the application-specific attribute definitions and the object classes are created during the installation or upgrade process. Application-specific attributes are qualified by an auxiliary object class, which will enable searching for the application-specific user properties of the entry. By default, application-specific entries are stored as `orclOwnerGUID=`*`GUID of the Base User`* under the `cn=User Properties, cn=`*`Application Type`*`, cn=Products,cn=OracleContext,` *`Realm DN`* container.

Some applications manage their own application attributes and implement the Data Access Java plug-in, which is described in "Understanding Provisioning Concepts" on

page 12-3. The Oracle Directory Integration Platform Service invokes this plug-in whenever the base user attributes or application-specific attributes are modified.

Figure 12–4 Base User and Application-Specific Attributes



This figure illustrates how base user and application-specific attributes are stored within the Oracle Internet Directory DIT. The `root` node contains two branches. The first branch represents the Oracle Identity Management realm and stores base user attributes, which include the user's status in provisioning-integrated applications, in the following DN: `cn=Users, dc=us, dc=MyCompany, dc=com`. In the illustration, this DN includes the following entries: `cn=user1` and `cn=user2`. The second branch contains three application containers. the DN for the first container is `cn=Application1, cn=Products, cn=OracleContext`, the DN for the second container is `cn=Application2, cn=Products, cn=OracleContext`, and the DN for the third container is `cn=Application3, cn=Products, cn=OracleContext`. The first and second application DNs contain `cn=user properties` containers that store application-specific attributes. The `cn=user properties` container for the first application DN contains a single entry, `orclOwnerGUID=user1GUID`. The `cn=user properties` container for the second application DN contains two entries: `orclOwnerGUID=user1GUID` and `orclOwnerGUID=user2GUID`. The third application DN contains callout text stating that the application-specific attributes are managed by the Data Access Java Plug-in.

****

12.5.2 Understanding User Provisioning Statuses

This section discusses the user provisioning statuses in Oracle Internet Directory. It contains these topics:

- Provisioning Status in Oracle Internet Directory
- Provisioning Status Transitions
- Upgrading and Coexistence Provisioning Statuses
- Provisioning Statuses and Exception Handling

12.5.2.1 Provisioning Status in Oracle Internet Directory

The Oracle Provisioning Service records a user's provisioning status in Oracle Internet Directory for each provisioning-integrated application. Provisioning status can be set by the Oracle Directory Integration Platform, with bulk provisioning using the provProfileBulkProv command, or by a provisioning-integrated application. Table 12–1 lists the provisioning statuses.

Table 12–1 Provisioning Statuses in Oracle Internet Directory

Internal Status	GUI Status	Description
Provisioning Statuses		
PROVISIONING_REQUIRED	Pending	Provisioning required. This status is selected by an administrator or set according to an application's provisioning policies. Note that this status determines whether a user has been provisioned.
PROVISIONING_IN_PROGRESS	In Progress	Provisioning in progress. The user can access the application when this is the current status if the application performs provisioning at scheduled intervals. The application can also provision the user on-demand.
PROVISIONING_SUCCESSFUL	Successful	Provisioning successful. This status is updated automatically by the Oracle Directory Integration Platform, with bulk provisioning using the provProfileBulkProv command, or a provisioning-integrated application.
PROVISIONING_NOT_REQUIRED	Not Requested	Provisioning not required. This status is selected by an administrator or set according to an application's provisioning policies. Note that this status determines whether a user will be provisioned.
PROVISIONING_FAILURE	Failed	Provisioning failed. This status is updated automatically by the Oracle Directory Integration Platform, with bulk provisioning using the provProfileBulkProv command, or a provisioning-integrated application. The user cannot access the application when this is the current status.
Deprovisioning Statuses		
DEPROVISIONING_REQUIRED	Pending de-provisioning	Deprovisioning required. The user is still provisioned when this is the current status.
DEPROVISIONING_IN_PROGRESS	De-provisioning In Progress	Deprovisioning in progress.
DEPROVISIONING_SUCCESSFUL	Successfully de-provisioned	Deprovisioning successful. The user cannot access the application when this is the current status.
DEPROVISIONING_FAILURE	Failed de-provisioning	Deprovisioning failed. The user is still provisioned when this is the current status.
Upgrade Statuses		
PENDING_UPGRADE	Pending Upgrade	Provisioning upgrade pending.
UPGRADE_IN_PROGRESS	Upgrade In Progress	Provisioning upgrade in progress.
UPGRADE_FAILURE	Upgrade Failed	Provisioning upgrade failed.

The provisioning status for each application is stored in the `orclUserApplnProvStatus` attribute in a user entry. This attribute is indexed in Oracle Internet Directory and is searchable. A subtyped `orclUserApplnProvStatus` attribute is created for each provisioning-integrated application. For example, the following statements store a user's provisioning statuses for an e-mail application and a scheduling application. The user's provisioning status for the e-mail application is `PROVISIONING_SUCCESS` while his or her provisioning status for the scheduling application is `PROVISIONING_FAILURE`.

```
orclUserApplnProvStatus;CORP-MAIL_E-MAIL:PROVISIONING_SUCCESS
orclUserApplnProvStatus;CORP-SCHEDULE_CALENDAR:PROVISIONING_FAILURE
```

Additional information about a user's provisioning status in an application is stored in the `orclUserApplnProvStatusDesc` attribute and the provisioning failure account for each application is stored in the `orclUserApplnProvFailureCount` attribute. As with the `orclUserApplnProvStatus` attribute, separate `orclUserApplnProvStatusDesc` and `orclUserApplnProvFailureCount` attributes are created for each provisioning-integrated application. The format for the `orclUserApplnProvStatusDesc` attribute is the same as the `orclUserApplnProvStatus` attribute, except that a timestamp and descriptive information are appended to the application name and type, as follows:

```
orclUserApplnProvStatusDesc;CORP-MAIL_E-MAIL:20040101010101^Missing employee ID
```

The `orclUserApplnProvStatus`, `orclUserApplnProvStatusDesc`, and `orclUserApplnProvFailureCount` attributes are contained in the `orclUserProvStatus` object class as optional attributes.

12.5.2.2 Provisioning Status Transitions

Table 12–2 lists the valid provisioning status transitions.

Table 12–2 Valid Provisioning Status Transitions in Oracle Internet Directory

Internal Status	GUI Status	Valid Transition From
Provisioning Statuses		
`PROVISIONING_REQUIRED`	Pending	*Initial missing state* `DEPROVISIONING_SUCCESSFUL`
`PROVISIONING_IN_PROGRESS`	In Progress	`PROVISIONING_REQUIRED`

`PROVISIONING_SUCCESSFUL`	Successful	`PROVISIONING_REQUIRED`
		`PROVISIONING_IN_PROGRESS`
		`PROVISIONING_FAILURE`

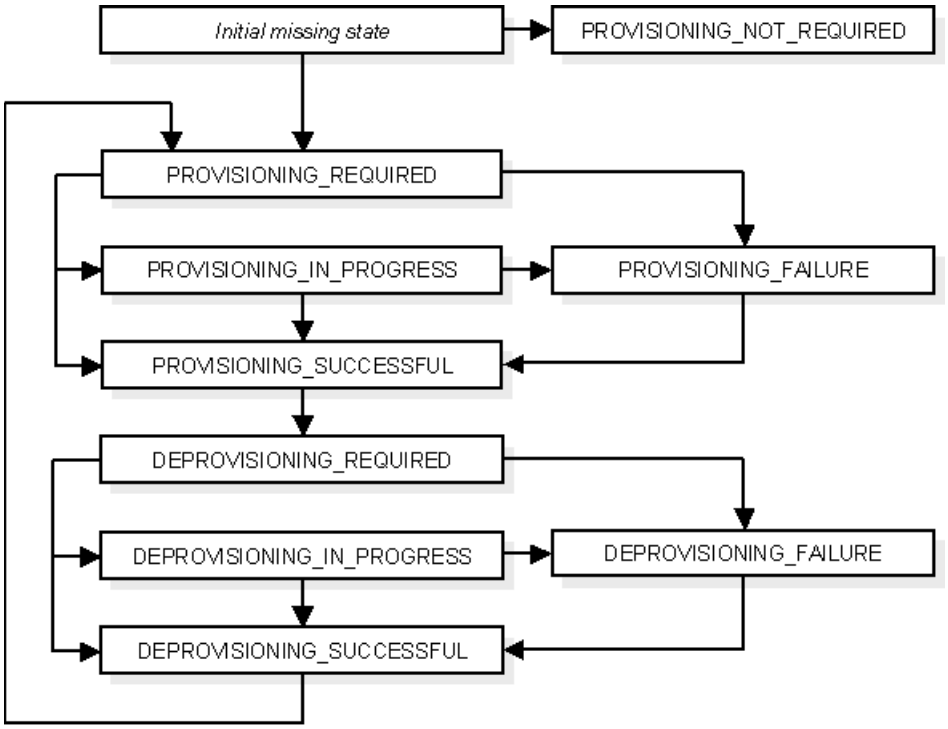
`PROVISIONING_NOT_REQUIRED`	Not Requested	*Initial missing state*
`PROVISIONING_FAILURE`	Failed	`PROVISIONING_REQUIRED` `PROVISIONING_IN_PROGRESS`
Deprovisioning Statuses		
`DEPROVISIONING_REQUIRED`	Pending de-provisioning	`PROVISIONING_SUCCESSFUL`

Table 12–2 (Cont.) Valid Provisioning Status Transitions in Oracle Internet Directory

Internal Status	GUI Status	Valid Transition From
DEPROVISIONING_IN_PROGRESS	De-provisioning In Progress	PROVISIONING_SUCCESSFUL
DEPROVISIONING_SUCCESSFUL	Successfully de-provisioned	DEPROVISIONING_REQUIRED
		DEPROVISIONING_IN_PROGRESS
		DEPROVISIONING_FAILURE
DEPROVISIONING_FAILURE	Failed de-provisioning	DEPROVISIONING_REQUIRED
		DEPROVISIONING_IN_PROGRESS

Figure 12–5 illustrates the valid provisioning status transitions.

Figure 12–5 Valid Provisioning Status Transitions



This figure conceptually illustrates the valid provisioning status transitions by using a series of boxes and arrows. See Table 12–2 for a listing of the valid provisioning status transitions.

****

12.5.2.3 Upgrading and Coexistence Provisioning Statuses

In Oracle Identity Management 11*g* Release 1 (11.1.1), a user entry can be physically represented in Oracle Internet Directory by multiple LDAP entries. In addition to the base user entry, separate LDAP entries can exist for each provisioning-integrated application.

In a typical upgrade of Oracle Identity Management, multiple middle tiers are not upgraded simultaneously. This means that following an Oracle Identity Management upgrade, middle tiers from a previous version may need to run in parallel with middle

tiers from the upgraded version. When a middle tier is upgraded, all of a user's application-specific data that was previously stored in the application metadata repository, will be migrated on-demand. For each user entry that is present in the Oracle Internet Directory back-end directory prior to the upgrade, the Oracle Directory Integration Platform will initiate a new user event and assign a provisioning status of `PENDING_UPGRADE` to the user entry. If a new user entry is created from an older middle tier or some unsupported route, such as an existing application using the standard LDAP SDK, the provisioning status attribute will be missing. In this case, the Oracle Directory Integration Platform also initiates a new user event and assign a provisioning status of `PENDING_UPGRADE` to the user entry.

Once a provisioning-integrated application receives the event, it will return a response to the Oracle Directory Integration Platform indicating whether or not the user is provisioned. The Oracle Directory Integration Platform then updates the provisioning status in the user entry accordingly.

12.5.2.4 Provisioning Statuses and Exception Handling

If a new user entry created with the user interface or through synchronization with an external data source does not contain enough information to provision the user in a particular application, provisioning may fail. Provisioning can also fail for a variety of other reasons. The Oracle Directory Integration Platform Service identifies user provisioning failures as exceptions.

Whenever an application responds to a `USER_ADD` event with a failure status, the Oracle Directory Integration Platform will change the user's provisioning status to `PROVISIONING_FAILURE`. The Oracle Directory Integration Platform will then send notifications to the applications of the failed cases also just like a new user case. This will serve as a retry for the provisioning request.

The provisioning status of a user displays in the user interface. The administrator can make the necessary changes to fix the problem, and the provisioning would get retried automatically. This will result in invocation of the data access plug-in if the provisioning is synchronous. However, an event will be propagated if the provisioning is asynchronous.

This sequence of steps will be retried as long as the user is not provisioned successfully.

12.6 Understanding Provisioning Flow

This section discusses the flow of information and control in various provisioning scenarios. It contains these topics:

- Viewing and Editing Provisioning Profiles Using Fusion Middleware Control
- User Provisioning from an External Source

12.6.1 Viewing and Editing Provisioning Profiles Using Fusion Middleware Control

As of 11*g* Release 1 (11.1.1), you view and edit provisioning profiles using the Oracle Enterprise Manager Fusion Middleware Control by performing the following steps:

1. Open a Web browser and enter the Oracle Enterprise Manager Fusion Middleware Control URL for your environment. The format of the Oracle Enterprise Manager Fusion Middleware Control URL is: https://*host*:*port*/em.
2. Log in to Oracle Enterprise Manager Fusion Middleware Control.

3. In the navigation panel on the left, click or expand the **Identity and Access** entry and then select the **DIP** component that contains the profile you want to view or edit.
4. Select **Administration** and then **Provisioning Profiles** from the DIP Server menu. The Manage Provision Profiles screen appears displaying the existing provisioning profiles.

   To change which attributes of the provisioning profiles are displayed, click **View**, then **Column**, and select the attributes you want to display or hide. You can also reorder the columns of provision profiles by clicking **View**, and then **Reorder Columns**.

   To enable or disable a provisioning profile, click the appropriate profile, and then click **Enable** or **Disable**.

   To edit a provisioning profile, click the profile you want to edit, and then click **Edit**. The attributes of the profile appear. Edit the settings as desired and click **OK** to save the changes. Table 12–3 lists and describes the provisioning profile fields:

Table 12–3 Provisioning Profile Fields

Field Name	Description
Profile Name	The name of the profile you are editing. You cannot edit a profile name after it is created. This field is provided only to identify the profile you are editing.
Application Name	The name of the application the provisioning profile applies to.
Profile Version	The version of the provisioning profile.
Application to OID	Options to set the provisioning profile as Configured and Enabled in the relationship between the Application and the Oracle back-end directory.
OID to Application	Options to set the provisioning profile as Configured and Enabled in the relationship between the Application and the Oracle back-end directory.
Scheduled Interval (HH:MM:SS)	Specifies the number of hours, minutes, and seconds between provisioning attempts between a connected directory and relationship between the application and the Oracle Internet Directory back-end directory.
Last Execution	Shows the status (Success/Failed) and execution time of the last provisioning attempt.

12.6.2 User Provisioning from an External Source

The majority of deployments are expected to provision users from an external source, such as a third-party enterprise user repository. In these types of deployments, the third-party repository bootstraps the Oracle Internet Directory back-end directory. Oracle Directory Integration Platform will provide ongoing synchronization between Oracle Internet Directory and the connected repository. Examples of connected user repositories include Human Resources and LDAP directories such as Microsoft Active Directory, Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server), Novell eDirectory, IBM Tivoli Directory Server, and OpenLDAP.

The Oracle Directory Synchronization Service will create the user entry in Oracle Internet Directory. Because the information coming from the external source may not be sufficient to provision the user in various applications, the application defaults will be used to create the application information.

User creation by the Oracle Directory Synchronization Service occurs as follows:

1. The Oracle Directory Synchronization Service evaluates the provisioning policies specified by the applications to determine whether the user should be provisioned in the application.
2. The Oracle Directory Synchronization Service evaluates any other plug-ins that the application has registered.
3. The Oracle Directory Integration Platform Service invokes the PL/SQL plug-in or the Data Access Java plug-in to deliver the user information to the application.
4. The provisioning status of the user is returned by the application using the event interfaces.
5. The Oracle Directory Integration Platform Service updates the provisioning status of the user for the application.

12.7 How Are Administrative Privileges Delegated?

Administrative rights in Oracle Delegated Administration Services vary according to the privileges delegated to each administrator. An administrator can be granted rights to manage and provision users, manage applications, or any combination of these privileges, as described in the following scenarios:

- Provisioning Administration Model

Note: Oracle Directory Integration Platform 11*g* Release 1 (11.1.1) interoperates with and supports Oracle Delegated Administration Services release 10.1.4.3.0 and higher.

12.7.1 Provisioning Administration Model

The following types of provisioning information is managed in the Oracle Internet Directory back-end directory:

- Base user information.
- Application-specific information.
- User provisioning status in each provisioning-integrated application; this information is stored in the base user entry but is administered separately.

Administrators and users each require the following types of privileges:

- Administrators require privileges for managing base user attributes and application-specific information.
- Users require privileges for managing their own base attributes and application-specific information.

User accounts with administrative privileges are represented by the group entry "`cn=User Provisioning Admins,cn=Groups,cn=OracleContext`". To manage application-specific information, the application must grant privileges to the "`cn=User Provisioning Admins,cn=Groups,cn=OracleContext`" group. If an application already defines a group with administrative privileges, then the application needs to add this group as a member of the group.

13

Deploying Provisioning-Integrated Applications

This chapter explains how to deploy provisioning-integrated applications with the Oracle Provisioning Service. It contains these topics:

- Deployment Overview for Provisioning-Integrated Applications
- Managing Provisioning Profiles Using oidprovtool
- Registering Applications for Provisioning
- Configuring Application Provisioning Properties

See Also:

- Chapter 4, "Managing the Oracle Directory Integration Platform"

13.1 Deployment Overview for Provisioning-Integrated Applications

To deploy provisioning-integrated applications with the Oracle Provisioning Service, you perform these general steps:

1. Install Oracle Internet Directory and Oracle Directory Integration Platform.
2. Load user information into Oracle Internet Directory.

   See Also: *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*

3. Start the Oracle Directory Integration Platform.
4. Install the applications and use the `oidprovtool` to create a provisioning profile for each application. Refer to "Managing Provisioning Profiles Using oidprovtool" on page 13-2 for more information.
5. Configure application registration by following the procedures described in "Registering Applications for Provisioning" on page 13-8.
6. Configure application provisioning by following the procedures described in "Configuring Application Provisioning Properties" on page 13-10.
7. Periodically monitor the status of the provisioning event propagation for each application. You can do this by using the Oracle Enterprise Manager Fusion Middleware Control.

See Also: The chapter on logging, auditing, and monitoring the directory in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*

13.2 Managing Provisioning Profiles Using oidprovtool

Provisioning enables you to ensure that an application is notified of directory changes, such as changes to user or group information. Such changes can affect whether the application allows a user access to its processes and resources.

When you install an application that you want to provision, you must create a provisioning integration profile using the `oidprovtool` command located in the *ORACLE_HOME*/bin directory.

You can use the `oidprovtool` to:

- Create a new provisioning profile. A new provisioning profile is created and set to the enabled state so that Oracle Directory Integration Platform can process it.
- Disable an existing provisioning profile.
- Enable a disabled provisioning profile.
- Modify an existing provisioning profile.
- Delete an existing provisioning profile.
- Get the current status of a given provisioning profile.
- Clear all of the errors in an existing provisioning profile.

The `oidprovtool` utility shields the location and schema details of the provisioning profile entries from the callers of the tool. From the callers' perspective, the combination of an application and a realm uniquely identify a provisioning profile. The constraint in the system is that there can be only one provisioning profile for each application for each realm.

Once a profile is created, its mode—that is, INBOUND, OUTBOUND, or BOTH—cannot be changed by using the `modify` operation. To change the mode, you must delete, then re-create, the profile.

The Oracle directory integration platform server automatically monitors provisioning profile configuration changes in Oracle Internet Directory, including the creation, modification, and deletion of provisioning profiles. For this reason, you do not need to manually enable or disable a provisioning profile.

Note: For improved security, do not enter a password with the `oidprovtool` command unless prompted for one.

13.2.1 Syntax for oidprovtool

oidprovtool

```
oidprovtool operation=[create|modify] ldap_host=oid_hostname ldap_port=port
ldap_user_dn="bindDN" ldap_user_password=password
[profile_mode=INBOUND|OUTBOUND|BOTH]
application_dn="DN" application_type=type [application_name=name]
[application_display_name=display name] organization_dn=DN
[application_isdasvisible=TRUE|FALSE] [manage_application_defaults=TRUE|FALSE]
[enable_bootstrap=TRUE|FALSE]  [user_data_location=DN]
[default_provisioning_policy=PROVISIONING_REQUIRED|PROVISIONING_NOT_REQUIRED]
```

```
interface_name=SCHEMA.PACKAGE [interface_type=PLSQL|JAVA]
interface_version=1.1|2.0|3.0] interface_connect_info=connection_string
schedule=number_seconds lastchangenumber=number
max_prov_failure_limit=number
max_events_per_schedule=number max_events_per_invocation=number
event_mapping_rules="OBJECT_TYPE:FILTER:DOMAIN"
event_permitted_operations="OBJECT:DOMAIN:OPERATION(attributes,...)"
event_subscription="USER|GROUP:DOMAIN:OPERATION(attributes,...)"
max_events_per_schedule=number max_retries=number profile_group=number
profile_status=ENABLED | DISABLED profile_debug=debug_level

oidprovtool {operation=enable|disable|delete|status|reset}
application_dn=DN [organization_dn=DN] [ldap_host=oid_hostname] [ldap_port=port]
[ldap_user_dn=bindDN] [ldap_user_password=password] [profile_debug=debug_level]
```

13.2.2 Arguments for oidprovtool

operation=create | modify | enable | disable | delete | status | reset

Required. The operation to perform using `oidprovtool`. You can only perform one operation at a time. The operations are:

- create—Creates a new provisioning profile.
- modify—Modifies the given properties of an existing provisioning profile.
- enable—Enables a provisioning profile.
- disable—Disables a provisioning profile.
- delete—Deletes a provisioning profile.
- status—Shows the current status of a given provisioning profile.
- reset—Clears all errors for a provisioning profile.

ldap_host=*oid_hostname*

Optional. The host name of the Oracle Internet Directory server. If not provided then the name of the local host is used.

ldap_port=*port*

Optional. The LDAP listening port of Oracle Internet Directory. The default is 389.

ldap_user_dn=*bindDN*

Required. The DN of the superuser or a user that has sufficient permissions to perform provisioning subscription operations. The default is `cn=orcladmin`.

ldap_user_password=*password*

Optional.The user password used to bind to the directory. If you do not specify the password on the command line, you will be prompted for it. Best security practice is to provide the password in response to a prompt.

profile_mode=OUTBOUND | INBOUND | BOTH

Optional for the `create` operation only. The direction of the provisioning events. The default is OUTBOUND (data is provisioned from Oracle Internet Directory to the application).

application_dn=*DN*

Required. The distinguished name of the application to which the provisioning subscription belongs. The combination of the application DN and organization DN uniquely identifies a provisioning profile. For example, here is the application DN for Portal:

```
"orclApplicationCommonName=PORTAL,cn=Portal,cn=Products,cn=OracleContext"
```

application_type=*type*

Required. The type of application being provisioned.

application_name=*name*

Optional. The name of the application being provisioned. If not provided, defaults to the distinguished name assigned to `application_dn`.

application_display_name=*name*

Optional. The display name of the application being provisioned. If not provided, defaults to the value assigned to `application_name`.

organization_dn=*DN*

Optional. If not provided, defaults to the default identity management realm. The distinguished name of the organization to which the provisioning subscription belongs, for example "`dc=company,dc=com`". The combination of the application DN and organization DN uniquely identifies a provisioning profile.

application_isdasvisible=TRUE | FALSE

Optional. Determines whether the application is visible as a provisioning-integrated application in the Oracle Internet Directory Provisioning Console. The default value is TRUE.

manage_application_default=TRUE | FALSE

Optional. Determines whether the Oracle Internet Directory Provisioning Console manages the application's default values. The default value is TRUE.

enable_bootstrap=TRUE | FALSE

Optional. Indicates whether the application should receive provisioning events for users that existed in Oracle Internet Directory before creating the application's provisioning integration profile. The default value is FALSE.

user_data_location=*DN*

Optional. Identifies the DN of the container in which to store application-specific user information.

default_provisioning_policy=PROVISIONING_REQUIRED | PROVISIONING_NOT_REQUIRED

Optional. Specifies the application's default provisioning policy. The default value is PROVISIONING_REQUIRED.

interface_name=SCHEMA.PACKAGE

Required for `create` or `modify` operations. The database schema name for the PLSQL package. The format of the value is `schema.package_name`, for example here is the schema and PLSQL package information for Portal:

```
interface_name=PORTAL.WWSEC_OID_SYNC
```

interface_version=1.1 | 2.0 | 3.0

The version of the interface protocol. Allowed values are 1.1, 2.0, or 3.0. The default value is 2.0.

interface_type=PLSQL | JAVA

Optional. The type of interface to which events will be propagated. The default is PLSQL.

interface_connect_info=*connection_string*

Required for `create` or `modify` operations. To connect to an Oracle database and propagate events, use one of the following formats for the connection string:

- DBURL=ldap://*ldaphost*:*ldapport*/*service*:*username*:*password* (recommended)
- *host*:*port*:*sid*:*username*:*password*
- DBSVC=*service*:*username*:*password*

schedule=*number_seconds*

Optional for `create` and `modify` operations only. The number of seconds between executions of this profile. The default is 3600, which means the profile is scheduled to be executed every hour.

lastchangenumber=*number*

Optional for `create` and `modify` operations on `OUTBOUND` events only. The last change number in Oracle Internet Directory after which all qualifying events should be provisioned to the application. Defaults to the latest current change number.

max_prov_failure_limit=*number*

Optional. Determines the number of times the Oracle Provisioning System attempts to provision a user. The default is 1.

max_events_per_schedule=*number*

Optional for `create` and `modify` operations only. The maximum number of events that the Oracle directory integration platform server sends to an application during one execution of a provisioning profile. The default is 100.

max_events_per_invocation=*number*

Optional for `create` and `modify` operations only. The maximum number of events that can be packaged and sent to a target in one invocation of the interface.

event_mapping_rules="*OBJECT_TYPE*:*FILTER*:*DOMAIN*"

Required for `create` and `modify` operations on `INBOUND` events only. This rule maps the object type received from the application (using an optional filter condition) to a domain in Oracle Internet Directory A provisioning profile can have multiple mapping rules defined.

The following example shows two mapping rules. The first rule shows that an employee object (`EMP`) whose locality attribute equals America (`l=AMERICA`) should be mapped to the domain `l=AMER,cn=users,dc=company,dc=com`. The second rule shows that an employee object (`EMP`) should be mapped to the domain `cn=users,dc=company,dc=com` (no filter conditions).

```
event_mapping_rules="EMP:l=AMERICA:l=AMER,cn=users,dc=company,dc=com"
event_mapping_rules="EMP::cn=users,dc=company,dc=com"
```

event_permitted_operations="*OBJECT*:*DOMAIN*:*OPERATION*(*attributes*,...)

Required for `create` and `modify` operations on `INBOUND` events only. This property is used to define the types of events that the application is allowed to send to the Oracle Directory Integration Platform service. A provisioning profile can have multiple permitted operations defined.

For example, if you wanted to permit the application to send events whenever a user object was added or deleted, or when certain attributes were modified, you would have three permitted operations such as this:

```
event_permitted_operations="USER:dc=mycompany,dc=com:ADD(*)"
event_permitted_operations="USER:dc=mycompany,dc=com:MODIFY(cn,sn,mail,password)"
event_permitted_operations="USER:dc=mycompany,dc=com:DELETE(*)"
```

event_subscription="USER | GROUP:*DOMAIN*:*OPERATION*(*attributes*,...)"

Required for `create` and `modify` operations on `OUTBOUND` events only. This property is used to define the types of events that the Oracle Directory Integration Platform service should send to the application. A provisioning profile can have multiple event subscriptions defined.

For example, if you wanted the directory integration server to send events to the application whenever a user or group object was added or deleted, you would have four event subscriptions such as this:

```
event_subscription="GROUP:dc=mycompany,dc=com:ADD(*)"
event_subscription="GROUP:dc=mycompany,dc=com:DELETE(*)"
event_subscription="USER:dc=mycompany,dc=com:ADD(*)"
event_subscription="USER:dc=mycompany,dc=com:DELETE(*)"
```

max_events_per_schedule=*number*

Optional for `create` and `modify` operations only. The maximum number of events to be provisioned in one schedule. The default is 100.

max_retries=*number*

Optional for `create` and `modify` operations only. The number of times a failed event should be retried. The default is 5.

profile_group=*number*

Required for `create` and `modify` operations only. The group number of the profile. Default is "DEFAULT". This is required to address scalability issues when different Oracle Directory Integration Platform server instances will be used to execute different selected groups.

profile_status=ENABLED | DISABLED

Required for the create operation only. Determines whether the profile is enabled or disabled. The default is ENABLED.

profile_debug=*debug_level*

Required. The debug level for the profile.

13.2.3 Tasks and Examples for oidprovtool

You can perform the following tasks using `oidprovtool`:

- Creating a Provisioning Profile
- Modifying a Provisioning Profile
- Deleting a Provisioning Profile
- Disabling a Provisioning Profile

13.2.3.1 Creating a Provisioning Profile

The following example creates a new provisioning profile that makes Portal aware of updates to the user and group information that is maintained in Oracle Internet Directory.

Example:

```
oidprovtool operation=create ldap_host=myhost.mycompany.com ldap_port=389 \
ldap_user_dn="cn=orcladmin" application_
dn="orclApplicationCommonName=PORTAL,cn=Portal,cn=Products,cn=OracleContext" \
organization_dn="dc=us,dc=mycompany,dc=com" interface_name=PORTAL.WWSEC_OID_SYNC \
interface_type=PLSQL interface_connect_info=myhost:1521:iasdb:PORTAL:password \
schedule=360 event_subscription="USER:dc=us,dc=mycompany,dc=com:DELETE" \
event_subscription="GROUP:dc=us,dc=mycompany,dc=com:DELETE" \
event_
subscription="USER:dc=us,dc=mycompany,dc=com:MODIFY(orclDefaultProfileGroup,userpa
ssword)" \
event_subscription="GROUP:dc=us,dc=mycompany,dc=com:MODIFY(uniqueMember)" \
profile_mode=OUTBOUND
```

13.2.3.2 Modifying a Provisioning Profile

The following example modifies an existing provisioning profile for the Portal application. It changes the event subscription for the attributes that are provisioned when a user entry is modified.

Example:

```
oidprovtool operation=modify ldap_host=myhost.mycompany.com ldap_port=389 \
ldap_user_dn="cn=orcladmin" application_
dn="orclApplicationCommonName=PORTAL,cn=Portal,cn=Products,cn=OracleContext" \
organization_dn="dc=us,dc=mycompany,dc=com" \
subscription="USER:dc=us,dc=mycompany,dc=com:MODIFY(orclDefaultProfileGroup,userpa
ssword,mail,cn,sn)"
```

13.2.3.3 Deleting a Provisioning Profile

The following example disables a provisioning profile for the Portal application.

Example:

```
oidprovtool operation=delete ldap_host=myhost.mycompany.com ldap_port=389 \
ldap_user_dn="cn=orcladmin" application_
dn="orclApplicationCommonName=PORTAL,cn=Portal,cn=Products,cn=OracleContext" \
organization_dn="dc=us,dc=mycompany,dc=com"
```

13.2.3.4 Disabling a Provisioning Profile

The following example disables a provisioning profile for the Portal application.

Example:

```
oidprovtool operation=disable ldap_host=myhost.mycompany.com ldap_port=389 \
ldap_user_dn="cn=orcladmin" application_
dn="orclApplicationCommonName=PORTAL,cn=Portal,cn=Products,cn=OracleContext" \
organization_dn="dc=us,dc=mycompany,dc=com"
```

13.3 Registering Applications for Provisioning

After you install an application and use the `oidprovtool` to create a provisioning profile for it, you must perform the following steps to register the application for provisioning:

1. Perform the initial provisioning registration and create a provisioning-integration profile. The Oracle Directory Integration Platform Service uses the provisioning-integration profiles to identify provisioning-integrated applications.
2. Provide the Oracle Directory Integration Platform Service with application-specific attributes, default values, and whether an attribute is mandatory when provisioning users for the application.
3. Register any plug-ins that are required by the provisioning-integrated application. This can include application-specific plug-ins that the application uses to enforce business policies.

> **Note:** The Oracle Directory Integration Platform Service does not support instance-level provisioning of applications that support a multiple instance architecture. If you install multiple instances of the same application, the Oracle Directory Integration Platform Service treats each instance as a separate provisioning-integrated application.

When creating users, an administrator can assign user attributes for a specific provisioning-integrated application. Application-specific attributes are stored in Oracle Internet Directory for each user that is provisioned for an application. For better performance, provisioning-integrated applications usually cache a local copy of user attributes instead of retrieving them from Oracle Internet Directory. Applications are notified of user creations, user deletions, and attribute modifications either synchronously with the Data Access Java plug-in or asynchronously with a PL/SQL plug-in.

Registration creates a unique identity for an application in Oracle Internet Directory. Oracle applications typically register themselves for provisioning by using the repository APIs located in the repository.jar file, which Oracle Application Server installs by default in the $*ORACLE_HOME*/jlib directory. In addition to creating an application entry in Oracle Internet Directory, the repository APIs can be used to add applications to privileged groups.

For non-Oracle applications that are not capable of using the registration APIs, you can use LDAP commands and LDIF templates to create identities for the applications in Oracle Internet Directory. You create a container for the application under `cn=Products,cn=OracleContext"` or `cn=Products, cn=OracleContext,` *`Realm DN`*. The container where you create an application identity depends on whether the application will be available to users in a single realm or multiple realms. In most cases, you should create an application identity in the `cn=Products, cn=OracleContext` container so the application is not bound by the identity

management policies of a specific Oracle Internet Directory identity management realm.

You can install multiple instances of the same application. Installing a new instance of a provisioning-integrated application creates a separate entry for the new instance under the application identity container. Although some configuration settings are instance-specific, other settings are shared across multiple instances of the same application. As an example, consider an application that is similar to Oracle Files. You can deploy multiple instances of Oracle Files in an environment where each instance is independent of other instances. You define each instance as a separate provisioning-integrated application. You can also provision users in multiple instances of the application.

When you install the first instance of an application, you must create in Oracle Internet Directory the entries shown in the following example. The example creates the application identity in the `cn=Products, cn=OracleContext` container, and assumes the application name and type are Files-App1 and FILES.

```
dn: cn=FILES,cn=Products,cn=OracleContext
changetype: add
objectclass: orclContainer
dn: orclApplicationCommonName=Files-App1,cn=FILES,cn=Products,cn=OracleContext
changetype: add
objectclass: orclApplicationEntity
orclappfullname: Files Application Instance 1
userpassword: password
description: This is a test application instance.
protocolInformation: protocol information
orclVersion: 1.0
orclaci: access to entry by group="cn=odisgroup,cn=DIPAdmins,cn=Directory
Integration Platform,cn=Products,cn=OracleContext" (browse,proxy) by
group="cn=User Provisioning Admins,cn=Groups,cn=OracleContext" (browse,proxy)
orclaci: access to attr=(*) by group="cn=odisgroup,cn=DIPAdmins,cn=Directory
Integration Platform,cn=Products,cn=OracleContext" (search,read,write,compare) by
group="cn=User Provisioning Admins,cn=Groups,cn=OracleContext"
(search,read,write,compare)
```

When you install the second instance of an application, you must create in Oracle Internet Directory the entries shown in the following example. The example also creates the application identity in the `cn=Products, cn=OracleContext` container, and assumes the application name is Files-App2.

```
dn: orclApplicationCommonName=Files-App2,cn=FILES,cn=Products,cn=OracleContext
changetype: add
objectclass: orclApplicationEntity
orclappfullname: Files Application Instance 2
userpassword: password
description: This is a test Appliction instance.
protocolInformation: protocol information
orclVersion: 1.0
orclaci: access to entry by group="cn=odisgroup,cn=DIPAdmins,cn=Directory
Integration Platform,cn=Products,cn=OracleContext" (browse,proxy) by
group="cn=User Provisioning Admins,cn=Groups,cn=OracleContext" (browse,proxy)
orclaci: access to attr=(*) by group="cn=odisgroup,cn=DIPAdmins,cn=Directory
Integration Platform,cn=Products,cn=OracleContext" (search,read,write,compare) by
group="cn=User Provisioning Admins,cn=Groups,cn=OracleContext"
(search,read,write,compare)
```

After you successfully register a provisioned-integrated application with Oracle Internet Directory, you may need to add the application to various privileged groups. Table 13–1 lists common privileged groups in Oracle Internet Directory.

Table 13–1 Common Privileged Groups in Oracle Internet Directory

Group	Description
`OracleDASCreateUser`	Create users
`OracleDASEditUser`	Edit users
`OracleDASDeleteUser`	Delete users
`OracleDASCreateGroup`	Create groups
`OracleDASEditGroup`	Edit groups
`OracleDASDeleteGroup`	Delete groups

The following LDIF file demonstrates how to grant create user privileges in all realms to the Files-App1 application:

```
dn:cn=OracleCreateUser,cn=Groups,cn=OracleContext
changetype: modify
add: uniquemember
uniquemember:
orclApplicationCommonName=Files-App1,cn=FILES,cn=Products,cn=OracleContext
```

13.4 Configuring Application Provisioning Properties

After you register a provisioning-integrated application, you must configure its properties. Each application's provisioning profile maintains its own provisioning configuration properties. Provisioning-integrated applications use properties to store the following types of metadata:

- Application identity information
- Identity realm information
- Default application provisioning policies
- Application attribute properties and defaults
- Application provisioning plug-ins
- Application event interface information
- Application event propagation information

Oracle Directory Integration Platform Provisioning supports three versions of provisioning profiles: 1.1, 2.0, and 3.0. Version 3.0 provisioning profiles are only available with Oracle Identity Management 11*g* Release 1 (11.1.1). Different applications support different provisioning profile versions. For example, many Oracle applications only support version 2.0. However, Oracle Collaboration Suite supports provisioning profile version 3.0. The primary differences between the provisioning profile versions are as follows:

- Provisioning applications that support provisioning profile versions 1.1 and 2.0 is a single-step process involving the `oidprovtool` utility, which is described in the chapter on Oracle Directory Integration Platform tools in the *Oracle Identity Management User Reference*. However, provisioning applications that support provisioning profile version 3.0 is a multiple-step process, which is described in

the centralized user provisioning Java API reference chapter of *Oracle Fusion Middleware Application Developer's Guide for Oracle Identity Management*.

- Oracle Directory Integration Platform Provisioning only maintains user provisioning status for applications that support provisioning profile version 3.0.

  See Also: The centralized user provisioning Java API reference chapter of *Oracle Fusion Middleware Application Developer's Guide for Oracle Identity Management*

14

Understanding the Oracle Provisioning Event Engine

This chapter discusses the Oracle provisioning event engine. It contains these topics:

- What Are the Oracle Provisioning Events?
- Working with the Oracle Provisioning Event Engine

14.1 What Are the Oracle Provisioning Events?

The Oracle provisioning event engine sends `USER_ADD`, `USER_MODIFY` and `USER_DELETE` events, depending on the operation performed on the user entries in Oracle Internet Directory. Because the user will be represented by multiple entries containing base user and application-specific user information, applications can subscribe to all of the attributes in the event.

The user events are also sent when a base entry or application entry is updated. However, no events are sent when an application entry is deleted because when an administrator requests the deprovisioning of a user from an application, a `USER_MODIFY` event is sent to the application with a provisioning status of `DEPROVISIONING_REQUIRED`. Once the application acknowledges the event by returning a value of `SUCCESS`, the application entry is deleted by the Oracle Directory Integration Platform.

To receive notification of provisioning status changes, an application must subscribe to the `orclUserApplnProvStatus;`*`Application_Name`* attribute. For example, to subscribe to the provisioning status change for an application named CORP_EMAIL, an application must subscribe to the `orclUserApplnProvStatus; CORP-EMAIL` attribute.

14.2 Working with the Oracle Provisioning Event Engine

The Oracle provisioning event engine generates events from add, modify, and delete operations that are performed on well-defined objects in Oracle Internet Directory. The Oracle provisioning event engine uses object definitions and event generation rules to generate events. This event generation model is extensible because it enables you to define custom objects and event generation rules. The Oracle provisioning event-engine, object definitions, and event generation rules are discussed in these topics:

- Creating Custom Event Object Definitions
- Defining Custom Event Generation Rules

14.2.1 Creating Custom Event Object Definitions

Table 14–1 lists the properties that you can use to identify objects for which events can be generated.

Table 14–1 Event Object Properties

Property	Description
`ObjectName`	Assigns a unique name to identify the object
`ObjectCriteria`	Identifies the LDAP object class to use for identifying the object
`MustAttributeCriteria`	Provides any additional attributes that are required for identifying the object
`OptionalAttributeCriteria`	Provides any optional attributes that may be required for identifying the object
`FilterAttributeCriteria`	Lists the attributes that should not be sent during event propagation

Table 14–2 lists the predefined objects for which the Oracle provisioning event engine can generate events.

Table 14–2 Predefined Event Objects

Object Name	Valid Object Class Values
`Entry`	`*`
`User`	`orclUserV2,inetorgperson`
`Identity`	`orclUserV2,inetOrgPerson`
`Group`	`groupOfUniqueNames,orclGroup, orclPrivilegeGroup,groupOfNames`
`Subscription`	`orclServiceSubscriptionDetail`
`Subscriber`	`orclSubscriber`

Note: The metadata for event objects is stored in the following container: `cn=Object Definitions, cn=Directory Integration Platform,cn=Products,cn=OracleContext`

14.2.2 Defining Custom Event Generation Rules

You specify event generation rules in XML format. The DTD for event generation rules is as follows:

```
<?xml version='1.0' ?>
  <!DOCTYPE EventRuleSet [
    <!ELEMENT ChangeType (#PCDATA)>
    <!ELEMENT Rule  (#PCDATA)>
    <!ELEMENT EventName  (#PCDATA)>
    <!ELEMENT ResEvent (Rule*, EventName)>
    <!ELEMENT EventRule (ChangeType, ResEvent*)>
    <!ELEMENT EventRuleSet (EventRule*) >
  ]>
```

The element definitions in the preceding DTD are as follows:

- The `EventRuleSet` root element identifies a set of event rules for an individual event object
- The `EventRuleSet` root element contains a list of `EventRule` elements
- Each `EventRule` element depends on the value assigned to the `ChangeType` element.
- The `ChangeType` and `Rule` elements determine the event name to be propagated to an application

Table 14–3 lists the event definitions that are supported by the Oracle provisioning event engine.

Table 14–3 Supported Event Definitions

Object Name	Change Type	Rule	Event Name
USER	Add	OrclApplnUserProvStatus=PENDING_UPGRADE	USER_ADD
	Add	OrclApplnUserProvStatus=PROVISIONING_REQUIRED	USER_ADD
	Modify	OrclApplnUserProvStatus= PENDING_UPGRADE	USER_ADD
		OrclApplnUserProvStatus=PROVISIONING_REQUIRED	USER_ADD
		OrclApplnUserProvStatus=PROVISIONING_FAILURE	USER_ADD
		OrclApplnUserProvStatus=DEPROVISIONING_REQUIRED	USER_MODIFY
		OrclApplnUserProvStatus=PROVISIONING_IN_PROGRESS	USER_MODIFY
		OrclApplnUserProvStatus=PROVISIONING_SUCCESSFUL	USER_MODIFY
	Delete	OrclApplnUserProvStatus=PROVISIONING_IN_PROGRESS	USER_DELETE
		OrclApplnUserProvStatus=PROVISIONING_SUCCESSFUL	USER_DELETE
		OrclApplnUserProvStatus=DEPROVISIONING_REQUIRED	
GROUP	Add		GROUP_ADD
	Modify		GROUP_MODIFY
	Delete		GROUP_DELETE
IDENTITY	Add		IDENTITY_ADD
	Modify		IDENTITY_MODIFY
	Delete		IDENTITY_DELETE
ENTRY	Add		ENTRY_ADD
	Modify		ENTRY_MODIFY
	Delete		ENTRY_DELETE
SUBSCRIPTION	Add		SUBSCRIPTION_ADD
	Modify		SUBSCRIPTION_ MODIFY
	Delete		SUBSCRIPTION_ DELETE

Table 14–3 (Cont.) Supported Event Definitions

Object Name	Change Type	Rule	Event Name
SUBSCRIBER	Add		SUBSCRIBER_ADD
	Modify		SUBSCRIBER_MODIFY
	Delete		SUBSCRIBER_DELETE

Note: The metadata for supported event objects is stored in the following container: `cn=Event Definitions, cn=Directory Integration Platform,cn=Products,cn=OracleContext`.

15

Integration of Provisioning Data with Oracle E-Business Suite

In Oracle Internet Directory 11*g* Release 1 (11.1.1), you can use the Oracle Directory Integration Platform Service to synchronize user accounts and other user information from Oracle E-Business Suite.

For information on how to use Oracle Directory Integration Platform Service to synchronize user accounts and other user information from Oracle E-Business Suite, refer to the following:

- Oracle E-Business Suite documentation. You can access Oracle E-Business Suite documentation on the Oracle Technology Network located at:

  `http://www.oracle.com/technology/index.html`

- The following Notes in My Oracle Support (formerly MetaLink), located at `http://metalink.oracle.com/`:
  - 233436.1—*Installing Oracle Application Server 10g with Oracle E-Business Suite Release 11i*
  - 261914.1—*Integrating Oracle E-Business Suite Release 11i with Oracle Internet Directory and Oracle Application Server Single Sign-On*
  - 233436.1—*Installing Oracle Application Server 10g with Oracle E-Business Suite Release 11i*

Part V

Integrating with Third-Party Directories

This part discusses the concepts, components, and procedures involved in integrating with various third-party identity directories. It contains these chapters:

16

Connected Directory Integration Concepts and Considerations

This chapter discusses the basic concepts of integrating Oracle Identity Management with a connected directory along with various decisions to be made as part of the integration process.

Note: This chapter assumes that you are familiar with:

- *Oracle Fusion Middleware Guide to Delegated Administration for Oracle Identity Management*.

This chapter contains these topics:

See Also: The following chapters for specific implementation details on synchronizing with connected directories:

16.1 Concepts and Architecture of Connected Directory Integration

Oracle provides centralized security administration for all Oracle components by integrating them with Oracle Identity Management. If your environment uses both an Oracle back-end directory and another directory, such as Microsoft Active Directory, you can use a connector to integrate the two systems and synchronize their data. A connector is a prepackaged connectivity solution that allows the Oracle back-end directory to synchronize with a connected directory.

This section discusses the Oracle components and architecture involved in integrating Oracle Identity Management with connected directories. It contains these topics:

- Oracle Identity Management Components for Integrating with Other Directories
- Oracle Back-end Directory Schema Elements for Synchronizing with Connected Directories
- Directory Information Tree in an Integration with a Connected Directory

---

Note: Refer to the *Oracle Identity Management Certification Information* for information about the directories and servers certified for integration with each of the Oracle back-end directories (Oracle Internet Directory, Oracle Unified Directory, and Oracle Directory Server Enterprise Edition).

You can access the *Oracle Identity Management Certification Information* from the Oracle Technology Network web site at:

`http://www.oracle.com/technology/index.html`

16.1.1 Oracle Identity Management Components for Integrating with Other Directories

This section describes the following components that are used to integrate Oracle Identity Management with another directory:

- The Oracle Back-end Directory
- Oracle Directory Integration Platform
- Oracle Delegated Administration Services
- Oracle Application Server Single Sign-On
- External Authentication Plug-ins

See Also: Chapter 3, "Administering Oracle Directory Integration Platform" for a description of the tools used to integrate Oracle Internet Directory with a third-party directory

The Oracle Back-end Directory

The Oracle back-end directory is the repository in which Oracle components and third-party applications store and access user identities and credentials. It uses an Oracle directory server to authenticate users by comparing the credentials entered by users with the credentials stored in the Oracle back-end directory. When credentials are stored in a connected directory and not in the Oracle back-end directory, users can still be authenticated if Oracle Internet Directory is the Oracle back-end directory. In this case, Oracle Internet Directory uses an external authentication plug-in that authenticates users against the connected directory server. (The external authentication plug-in is not available if Oracle Unified Directory or Oracle Directory Server Enterprise Edition is the Oracle back-end directory.)

See Also: If Oracle Internet Directory is the Oracle back-end directory, see the chapter on security in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for a discussion about security.

Oracle Directory Integration Platform

Oracle Directory Integration Platform is installed as part of Oracle Identity Management. You can configure it to run on the same host as the Oracle back-end directory or on a different host.

Oracle Directory Integration Platform enables:

- Synchronization between the Oracle back-end directory (either Oracle Internet Directory, Oracle Unified Directory, or Oracle Directory Server Enterprise Edition) and other connected directories and user repositories.
- Automatic provisioning services for Oracle components if Oracle Internet Directory is the Oracle back-end directory.

Oracle Directory Integration Platform includes connectors to synchronize the Oracle back-end directory with other LDAP directories or data stores. The Oracle Directory Integration Platform integration connectors allow you to:

- Configure either one-way or two-way synchronization with a connected directory.

  Note: Two-way password synchronization is only supported if the back-end directory is Oracle Internet Directory.

  Oracle Directory Integration Platform does not support password synchronization to connected directories from an Oracle Unified Directory back-end directory or an Oracle Directory Server Enterprise Edition back-end directory.

  One-way password synchronization from connected directories to either an Oracle Unified Directory back-end directory or an Oracle Directory Server Enterprise Edition back-end directory is supported.

- Designate a specific subset of attributes for synchronization. You do this by configuring the appropriate mapping rules, which you can then change at run time.

  See Also: "Attribute-Level Mapping" on page 6-6 for a discussion about configuring attribute mapping rules

Oracle Delegated Administration Services

Oracle Delegated Administration Services is a set of pre-defined, Web-based units for performing directory operations on behalf of a user. It frees directory administrators from the more routine directory management tasks by enabling them to delegate specific functions to other administrators and to end users. It provides most of the functionality that directory-enabled applications require, such as creating a user entry, creating a group entry, searching for entries, and changing user passwords. To administer application data in the directory, you use the Oracle Internet Directory Self-Service Console, a tool based on Oracle Delegated Administration Services. This tool comes ready to use with Oracle Internet Directory. Or, you can use Oracle Delegated Administration Services to develop your own tools for administering application data.

> **See Also:** *Oracle Fusion Middleware Guide to Delegated Administration for Oracle Identity Management*

Oracle Application Server Single Sign-On

OracleAS Single Sign-On Server enables users to access Oracle Web-based components by logging in only once.

> **Note:** Your back-end directorymust be Oracle Internet Directory to use OracleAS Single Sign-On Server. Single sign-on is not supported if either Oracle Unified Directory or Oracle Directory Server Enterprise Edition is your back-end directory.

Oracle components delegate the login function to the OracleAS Single Sign-On Server. When a user first logs in to an Oracle component, the component directs the login to the OracleAS Single Sign-On Server. The OracleAS Single Sign-On Server compares the credentials entered by the user to those stored in Oracle Internet Directory. After verifying the credentials, the OracleAS Single Sign-On Server grants the user access to all components the user is authorized to use throughout the current session.

Oracle Application Server Single Sign-On enables native authentication in a Microsoft Windows environment. Once logged in to the Windows environment, the user automatically has access to Oracle components. OracleAS Single Sign-On Server automatically logs in the user to the Oracle environment using the user's Kerberos credentials.

> **See Also:** *Oracle Enterprise Single Sign-On Suite Plus Administrator's Guide* for information about OracleAS Single Sign-On Server

External Authentication Plug-ins

External authentication plug-ins, such as the Microsoft Active Directory external authentication plug-in, are available for Oracle Internet Directory and enable users to log in to the Oracle environment by using their Microsoft Windows credentials. (External authentication plug-ins are not available for Oracle Unified Directory or Oracle Directory Server Enterprise Edition.) When an external authentication plug-in is in place, it is invoked by the Oracle directory server. This plug-in verifies the user's credentials in a connected directory. If the verification is successful, then the Oracle directory server notifies OracleAS Single Sign-On Server.

16.1.2 Oracle Back-end Directory Schema Elements for Synchronizing with Connected Directories

The Oracle back-end directory contains schema elements that correspond to attributes that are specific to connected directories, such as Microsoft Active Directory. The schema elements identify back-end directory objects that Oracle Directory Integration Platform synchronizes with the connected directory.

These schema elements are described later in the chapter in the following sections:

- Section 16.3.4, "Oracle Back-end Directory Schema Elements for Microsoft Active Directory"
- Section 16.4.2, "Oracle Internet Directory Schema Elements for Oracle Directory Server Enterprise Edition (Sun Java System Directory Server)"
- Section 16.5.2, "Oracle Back-end Directory Schema Elements for IBM Tivoli Directory Server"

- Section 16.6.2, "Oracle Back-end Directory Schema Elements for Novell eDirectory"
- Section 16.6.3, "Oracle Back-end Directory Schema Elements for OpenLDAP"

16.1.3 Directory Information Tree in an Integration with a Connected Directory

This section contains these topics:

- About Realms in Oracle Internet Directory
- Planning the Deployment
- Example: Integration with a Single Connected Directory Domain

See Also:

- The chapter on directory concepts and architecture in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for a fuller discussion of directory information trees
- The chapter in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* about the deployment of identity management realms

16.1.3.1 About Realms in Oracle Internet Directory

In Oracle Internet Directory, an identity management realm defines an enterprise scope over which certain identity management policies are defined and enforced by the deployment.

An identity management realm comprises:

- A well-scoped collection of enterprise identities—for example, all employees in the US domain.
- A collection of identity management policies associated with these identities. An example of an identity management policy would be to require that all user passwords have at least one alphanumeric character.
- A collection of groups, that is, aggregations of identities that simplify setting the identity management policies

Multiple Realms

You can define multiple identity management realms within the same Oracle Identity Management infrastructure. This enables you to isolate user populations and enforce a different identity management policy,—for example, password policy, naming policy, self-modification policy—in each realm. This is useful in a hosted deployment of Oracle Fusion Middleware.

Each identity management realm is uniquely named to distinguish it from other realms. It also has a realm-specific administrator with complete administrative control over the realm.

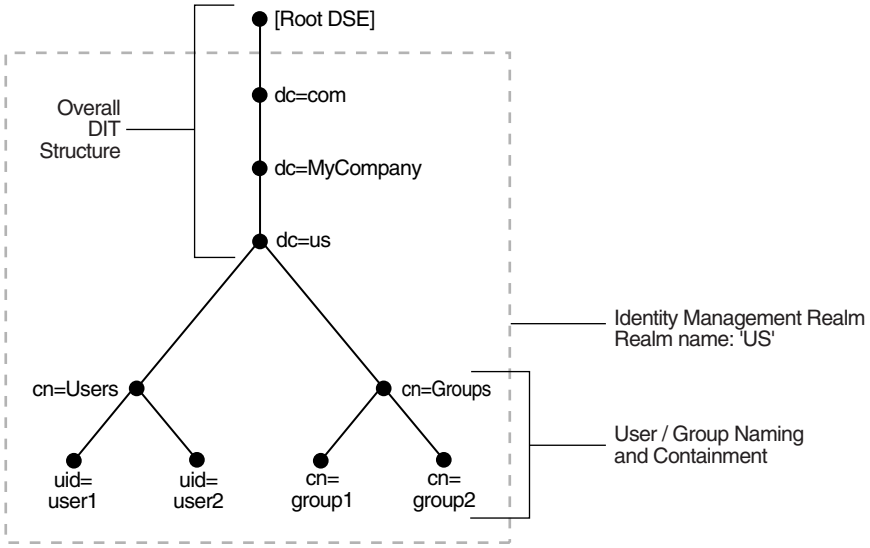
The Default Realm

For all Oracle components to function, an identity management realm is required. One particular realm, created during installation of Oracle Internet Directory, is called the default identity management realm. It is where Oracle components expect to find users, groups, and associated policies whenever the name of a realm is not specified. This default realm facilitates proper organization of information and enforces proper access controls in the directory.

There can be only one default identity management realm in the directory. If a deployment requires multiple identity management realms, then one of them must be chosen as the default.

Figure 16–1 illustrates the default identity management realm.

Figure 16–1 The Default Identity Management Realm



As Figure 16–1 shows, the default identity management realm is part of a global DIT. The node that follows the root DSE is `dc=com`, followed by `dc=MyCompany`, then `dc=us`. These four nodes represent the overall DIT structure. The node `dc=us` is the root of the default identity management realm. It has two subtrees for containing user and group information: `cn=Users` and `cn=Groups`. For illustration purposes, the `cn=Users` node contains two leaves: `uid=user1` and `uid=user2`. Similarly, the `cn=Groups` node contains `cn=group1` and `cn=group2`.

Access Control Policies in the Realm

You must configure appropriate ACLs in Oracle Internet Directory to enable Oracle Directory Integration Platform to:

- Enable the import profile to add, modify and delete objects in the `users` and `groups` containers. By default, import profiles are part of the Realm Administrators group, which can perform all operations on any entry under the realm DN. If you have customized ACLs in the realm, then be sure that the import profiles have the appropriate privileges to perform these operations on the subtree to be synchronized or on either the `user` container, the `group` container, or both depending on where the synchronization takes place.
- Enable Oracle components to manage the users and groups in the realm. By default, Oracle components can manage users and groups in the `users` and `groups` containers respectively. If you have updated your `usersearchbase` and `groupsearchbase` in the realm, then set up appropriate ACLs on the `users` container and `groups` container.

> **See Also:** The chapter on deployment of Oracle Identity Management realms in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for a description of the default realm installed with Oracle Internet Directory

16.1.3.2 Planning the Deployment

When planning the DIT, the most important decisions to make before synchronization are:

- Which directory is to be the central one
- What objects to synchronize, for example:
  - The portion of the DIT that you want to synchronize. You can synchronize the entire DIT or just a portion of it.
  - For each entry, the specific contents that you want to synchronize. You can synchronize the entire content of the entry or just a portion of it.
- Where to synchronize. You have two options:
  - You can synchronize so that the relative position of each entry in the DIT is the same in the source and destination directories. This configuration, called one-to-one distinguished name mapping, is the most commonly used configuration. Because the source DN is the same as the destination DN, this configuration provides better performance than when the two DNs are different.
  - You can synchronize so that the relative position in the DIT of each entry in the destination directory is different from that in the source directory. In this configuration, the Oracle Directory Integration Platform must change the DN values of all entries being mapped, including their references in group entries. This requires more intensive computation.

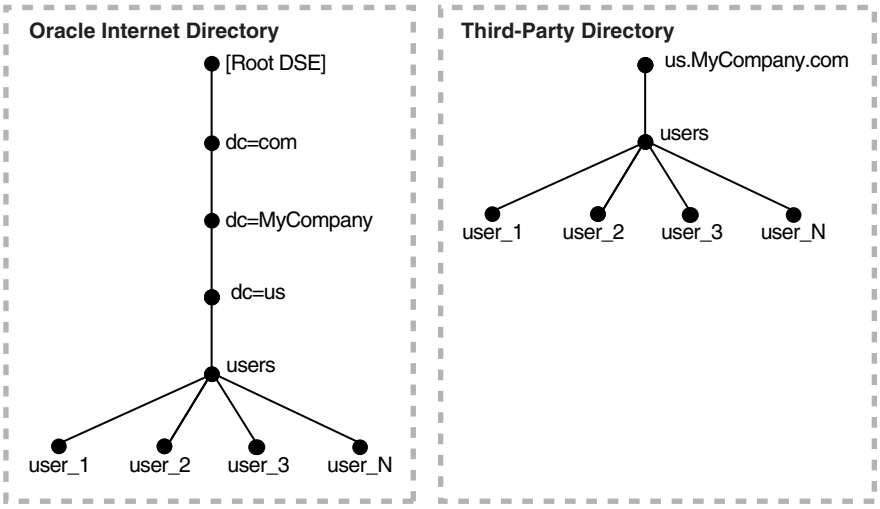
    If you synchronize in this way, you need to use the `dnconvert` mapping rule as described in "Supported Attribute Mapping Rules and Examples" on page 6-11.

    See Also: The section "Choose the Structure of the Directory Information Tree" on page 16-16 for more information about planning the directory information tree

16.1.3.3 Example: Integration with a Single Connected Directory Domain

Figure 16–2 shows an example of one-to-one mapping between Oracle Internet Directory and a connected directory.

Figure 16–2 Default DIT Structures in Oracle Internet Directory and a Connected Directory When Both Directory Hosts Are Under the Domain us.MyCompany.com



This illustration shows a directory information tree (DIT) in Oracle Internet Directory and a corresponding single domain in a connected directory. The DIT in Oracle Internet Directory extends downward from the root DSE to `dc=com` to `dc=MyCompany` to `dc=us` to `users`. From there it extends to four leaves: `user_1`, `user_2`, `user_3`, and `user_4`. The single domain in the connected directory has `us.MyCompany.com` as the root. From there it extends downward to users, and from there to four leaves: `user_1`, `user_2`, `user_3`, and `user_4`.

****

In the one-to-one mapping illustrated in Figure 16–2:

- Both Oracle Internet Directory and the connected directory hosts have the same topology.
- Users are synchronized only from the connected directory to Oracle Internet Directory. All users to be synchronized are stored in one container in the connected directory, in this case `users.us.MyCompany.com`.
- The same DIT structure is maintained in both the connected directory and Oracle Internet Directory. All users appear in the same users subtree identified by the value `cn=users,dc=us,dc=MyCompany,dc=com`.

In the example shown in Figure 16–2, only the `users` subtree must be synchronized from the connected directory to Oracle Internet Directory using one-to-one domain mappings.

Note: In Figure 16–2, the two directories have the same topology, but be aware that this is for illustration purposes only. The two directories do not need to be in the same domain. Oracle Internet Directory can be anywhere in the network, provided it can connect to the connected directory.

In addition, although the synchronization in the example is one-way, from the connected directory to Oracle Internet Directory, the synchronization can, alternatively, be bi-directional.

16.2 Planning Your Integration Environment

This section describes how to plan your integration environment. It contains these topics:

16.2.1 Preliminary Considerations for Integrating with a Connected Directory

If you are deploying an Oracle back-end directory in an enterprise that already has an LDAP directory server, then you must configure both directories to coexist in the same environment.

The coexistence of directories requires either of two different types of deployments:

- Simple synchronization with the Oracle back-end directory. Use this approach if your environment supports enterprise users by using a database server. If Oracle Internet Directory is your Oracle back-end directory, this approach will support Enterprise User Security.

  Note: Your Oracle back-end directory must be Oracle Internet Directory to support Enterprise User Security. The Oracle Unified Directory back-end directory and the Oracle Directory Server Enterprise Edition back-end directory do not support integration with other Fusion Middleware components, including Enterprise User Security.

- Complete integration with the Oracle Fusion Middleware infrastructure. This enables all enterprise users to use the various components in the Oracle Fusion Middleware suite. Use this approach if your environment uses a connected directory as the enterprise directory and deploys an Oracle Fusion Middleware suite of applications.

  Because all Oracle Fusion Middleware components depend on the identity management realm, complete integration with the Oracle Fusion Middleware infrastructure requires you to make some decisions about the container for that realm. Once you have made these decisions, you can configure bootstrapping and synchronization between the directories.

16.2.2 Choose the Directory for the Central Enterprise Directory

This section explains how to choose which directory is to be the central enterprise directory or *metadirectory*. It contains these topics:

- Scenario 1: Oracle Internet Directory as the Central Enterprise Directory
- Scenario 2: A Directory Other Than Oracle Internet Directory is the Central Enterprise Directory

16.2.2.1 Scenario 1: Oracle Internet Directory as the Central Enterprise Directory

If Oracle Internet Directory is the central directory, then, once the user, group, and realm objects are created, Oracle Internet Directory becomes the source of provisioning information for all Oracle components and connected directories. The user and group objects for the entire enterprise are then provisioned in various Oracle components and connected directories from Oracle Internet Directory.

> **Note:** In scenario one, Oracle Unified Directory or Oracle Directory Server Enterprise Edition can also serve as the central enterprise directory. Only Oracle Internet Directory, however, supports Oracle Application Server Single Sign-on.

Table 16–1 describes the typical requirements in this deployment.

Table 16–1 Typical Requirements with Oracle Internet Directory as the Central Enterprise Directory

Requirement	Description
Initial startup	The syncProfileBootstrap command populates the connected directory with users and groups stored in Oracle Internet Directory.
Synchronization	User and group information is managed in Oracle Internet Directory. Changes to that information are synchronized with the connected directory by Oracle Directory Integration Platform when an export profile has been configured. Synchronization from the connected directory into Oracle Internet Directory can be achieved by configuring an import profile.
Passwords and password verifiers	Passwords are managed in Oracle Internet Directory by using Oracle tools such as the Oracle Internet Directory Self-Service Console. Password changes are synchronized with the connected directory by the Oracle Directory Integration Platform. However, before this server can synchronize the password changes, the password synchronization must be configured in the mapping rules. Because the password is securely managed, the communication for synchronizing passwords to the connected directory must be over SSL. Run the Oracle Directory Integration Platform in the server authentication mode with the proper certificate from the connected directory. Be sure that the connected directory is also enabled for SSL. If the Oracle environment requires a password verifier, then the password verifier is automatically generated when a new user entry is created or when a password is modified.
Oracle Application Server Single Sign-On (version 10.1.4.*x*)	Users log in to the Oracle environment by using the OracleAS Single Sign-On Server. When called upon by the OracleAS Single Sign-On Server to authenticate a user, the Oracle directory server uses credentials available locally. No external authentication is involved. Users must log in only once to access various components in the Oracle environment.

New users or groups in Oracle Internet Directory can be automatically provisioned by the Oracle Directory Integration Platform. This automatic provisioning requires that:

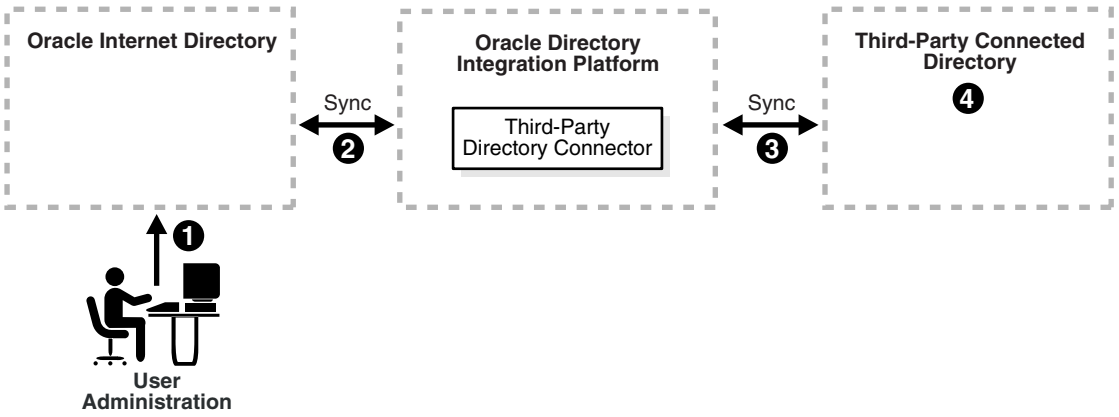
- The Oracle directory server is running with the change log enabled
- The change log is not purged

If these two conditions are not met, then you must dump the entries in Oracle Internet Directory to an LDIF file and upload the data to the connected directory.

> **See Also:** The chapter on garbage collection in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about purging the change log

Figure 16–3 shows a typical deployment in which Oracle Internet Directory is the central enterprise directory.

Figure 16–3 Interaction Among Components with Oracle Internet Directory as the Central Enterprise Directory



As Figure 16–3 on page 16-11 shows, when Oracle Internet Directory is the central enterprise directory, typical provisioning of a user or group follows this process:

1. The user or group entry is created in Oracle Internet Directory by using the Oracle Internet Directory Self-Service Console or command-line tools.
2. At the next scheduled interval, that entry creation event is read by the third-party directory connector in Oracle Directory Integration Platform.
3. Following the mapping information in the integration profile, the user or group attributes in Oracle Internet Directory are appropriately mapped to the corresponding user or group attributes as required by the schema in the connected directory.
4. The user and group entry is created in the connected directory.

A user entry is modified in Oracle Internet Directory, when:

- A new attribute gets added to the entry.
- The value of an existing attribute is modified.
- An existing attribute is deleted.

When Oracle Internet Directory is the central enterprise directory, the sequence of events during modification of a user or group entry is as follows:

1. The entry is modified by using the Oracle Internet Directory Self-Service Console or Oracle Enterprise Manager Fusion Middleware Control.

2. At the next scheduled interval, that entry modification event is read by the third-party directory connector in Oracle Directory Integration Platform.
3. Following the mapping information in the integration profile, the attribute in Oracle Internet Directory is appropriately mapped to the corresponding attribute in the connected directory.
4. The user entry is modified in the connected directory.

16.2.2.2 Scenario 2: A Directory Other Than Oracle Internet Directory is the Central Enterprise Directory

In this scenario either a third-party directory or another Oracle directory, such as Oracle Unified Directory or Oracle Directory Server Enterprise Edition, is the central enterprise directory, and Oracle Internet Directory is the Oracle back-end directory. In this scenario, once the user, group, and realm objects are created, the central enterprise directory becomes the source of provisioning information for all Oracle components and other directories. Oracle Internet Directory is deployed as the Oracle back-end directory to support Oracle components. To provide this support, Oracle Internet Directory stores a footprint that enables it to identify entries in the connected directory.

Table 16–2 describes the typical requirements in this deployment.

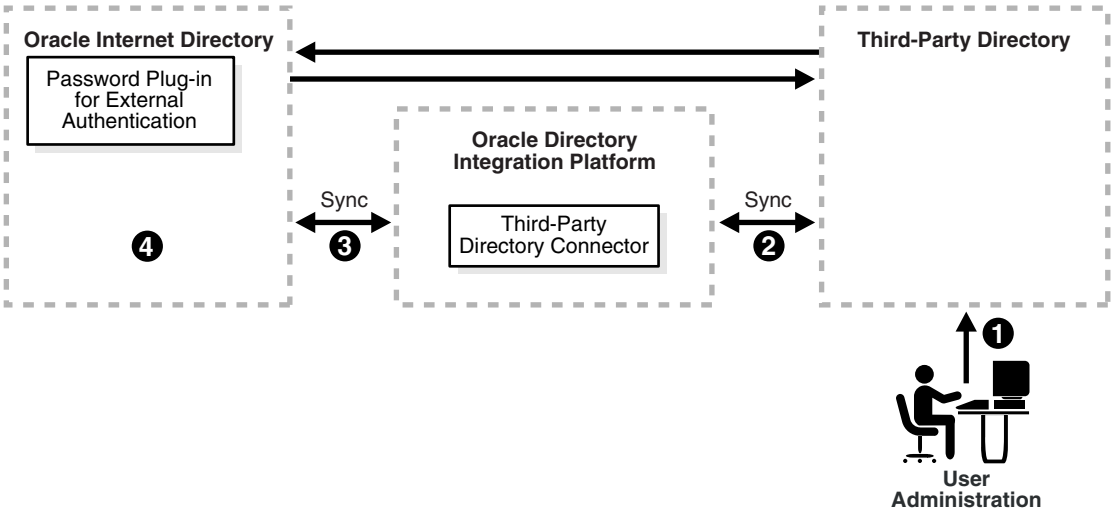
Table 16–2 Typical Requirements if a Directory Other Than Oracle Internet Directory is the Central Enterprise Directory, but Oracle Internet Directory is the Back-end Directory

Requirement	Description
Initial startup	The syncProfileBootstrap command populates Oracle Internet Directory with users and groups stored in the central enterprise directory.
	You can choose to manage user information, including password credentials, in the central enterprise directory only. In such deployments, to enable single sign-on in the Oracle environment, the Oracle Directory Integration Platform can synchronize only those user entry attributes required by Oracle components.
	Passwords are not migrated from the central enterprise directory to Oracle Internet Directory.
Synchronization	The central enterprise directory for user and group information is Oracle Unified Directory, Oracle Directory Server Enterprise Edition, or a third-party directory. Changes to user and group information in the central enterprise directory are synchronized with Oracle Internet Directory by the Oracle Directory Integration Platform when an import profile has been configured.
	Synchronization from Oracle Internet Directory to the central enterprise directory is achieved by configuring an export profile.
Passwords and password verifiers	Passwords are managed in the central enterprise directory. The Oracle Directory Integration Platform does not synchronize password changes into Oracle Internet Directory.
Oracle Application Server Single Sign-On	Users log in to the Oracle environment only once by using the OracleAS Single Sign-On Server.
	Users with credentials only in the central enterprise directory are authenticated by Oracle Internet Directory invoking the external authentication plug-in.
	Users with credentials in Oracle Internet Directory are authenticated locally.
Third-party directory external authentication plug-in	When user credentials are managed in Oracle Unified Directory, Oracle Directory Server Enterprise Edition, or a third-party directory, Oracle Internet Directory requires this plug-in. To authenticate a user, the OracleAS Single Sign-On Server calls upon the Oracle Internet Directory directory server. The plug-in then performs the authentication of the user against the user credentials stored in Oracle Unified Directory, Oracle Directory Server Enterprise Edition, or the third-party directory.

New users or groups created in the directory that is designated as the central enterprise directory are automatically synchronized into Oracle Internet Directory by the Oracle Directory Integration Platform. Before the provisioning can take place, a one-way synchronization between the central enterprise directory and Oracle Internet Directory must be established.

Figure 16–4 shows a typical deployment where a third-party directory is the central enterprise directory.

Figure 16–4 Interaction of Components with a Third-Party Directory as the Central Enterprise Directory



16.2.2.2.1 Process for Provisioning of a User or Group Figure 16–4 shows the typical process for provisioning a user or group when a third-party directory is the central enterprise directory.

> **Note:** Oracle Unified Directory and Oracle Directory Server Enterprise Edition do not support provisioning.

This process is described as follows:

1. The user or group entry is created in the third-party directory.
2. At the next scheduled interval, the entry creation event is read by the third-party directory connector in Oracle Directory Integration Platform.
3. Following the mapping information in the integration profile, the user or group attributes in the third-party directory are mapped to the corresponding attributes in the third-party directory.
4. The user or group entry is created in Oracle Internet Directory.

16.2.2.2.2 Process for Modifying a User or Group Entry An entry is modified in the connected directory when:

- A new attribute gets added to the entry.
- The value of an existing attribute is modified.
- An existing attribute is deleted.

When a connected directory is the central enterprise directory, modification of a user or group entry follows this process:

1. The entry is modified in the connected directory.
2. At the next scheduled interval, that entry modification event is read by the third-party directory connector in Oracle Directory Integration Platform.
3. Following the mapping information in the integration profile, the attribute in the connected directory is appropriately mapped to the corresponding attribute in Oracle Internet Directory.
4. The user or group entry is modified in Oracle Internet Directory.

As Figure 16–4 shows, when a third-party directory is the central enterprise directory, modification of passwords happens asynchronously in the directory that serves as the password repository. This happens by using plug-ins.

16.2.3 Customizing the LDAP Schema

Customizing the LDAP schema is required if:

- A directory deployment contains schema extensions such as custom object classes and attributes
- The custom attributes must be synchronized from one directory server to the other

To customize the LDAP schema, you must:

- Identify the schema extensions on the source directory
- Create those extensions on the target directory before starting the data migration and the synchronization

---

Note: In addition to creating schema extensions, you must also add the attribute to be synchronized with the corresponding object classes to the mapping rules.

See Also:

- The chapter on administering the schema in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for instructions on customizing the schema in Oracle Internet Directory
- Microsoft documentation available at `http://msdn.microsoft.com/` for instructions on customizing the schema in Microsoft Active Directory

16.2.4 Choose Where to Store Passwords

Regardless of which directory is the central enterprise directory, the password can be stored in one or both directories. There are advantages and disadvantages to each option. This section compares the two options in these topics:

- Advantages and Disadvantages of Storing the Password in One Directory
- Advantages and Disadvantages of Storing Passwords in Both Directories

16.2.4.1 Advantages and Disadvantages of Storing the Password in One Directory

Storing the password in one directory can make the password more secure because it reduces the number of points of entry. Further, it eliminates synchronization issues when the password is modified.

On the other hand, storing the password in one directory provides a single point of failure for the entire network. If the connected directory fails, then even though user footprints are available in Oracle Internet Directory, users cannot access Oracle components.

Although storing passwords in the central directory eliminates possible synchronization issues, it requires you to enable applications to authenticate users to that directory. This involves using the appropriate plug-ins. For example, if you are using Microsoft Active Directory as both the central enterprise directory and the central password store, then you must enable applications to authenticate users to Microsoft Active Directory. You do this by using an external authentication plug-in.

Note: Oracle components use password verifiers to authenticate users, and, when passwords are stored in a third-party directory, those verifiers are not stored in Oracle Internet Directory. If a password is modified by using an Oracle component, then the verifiers are both generated and stored in Oracle Internet Directory.

16.2.4.2 Advantages and Disadvantages of Storing Passwords in Both Directories

If you decide to store passwords in both Oracle Internet Directory and a connected directory, then passwords need to be synchronized, ideally in real-time.

In Oracle Internet Directory 11*g* Release 1 (11.1.1), passwords are not synchronized in real time, but according to a schedule. This can mean an observable delay between the time the password is changed in the central enterprise directory and the time that the change is recorded in the other directory.

In deployments with Oracle Internet Directory as the central directory, password values are synchronized regularly from Oracle Internet Directory to the connected directory. This requires you to enable both the password policy of the realm and reversible encryption.

See Also:

- The chapter in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* about password policies for information about setting password policies
- The chapter in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* about directory storage of password verifiers for information about reversible encryption

In general, password values are hashed. If both directories use the same hashing algorithm, then the hashed values can be synchronized as they are. For example, suppose that you have an environment in which Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server) and Oracle Internet Directory are integrated. Both of these directories support common hashing algorithms. If the passwords are hashed and stored in Oracle Directory Server Enterprise Edition by using a hashing technique supported by Oracle Internet Directory, then synchronizing Oracle Directory Server Enterprise Edition passwords to Oracle Internet Directory is the same as with any other attribute. If both directories do not support the same

hashing algorithm, then passwords must be synchronized in clear text format only. For security reasons, password synchronization is possible with Oracle Internet Directory only in SSL server authentication mode.

If Oracle Internet Directory is the central directory, and if the hashing algorithm it supports is not supported by the other directory, then synchronization is still possible through SSL server authentication mode when reversible password encryption is enabled.

If Microsoft Active Directory is the central directory, then, when a password is modified in Microsoft Active Directory, a plug-in intercepts the password changes and sends them to Oracle Internet Directory. When Oracle Internet Directory is the central directory and the central password store, Oracle Directory Integration Platform reads the password changes as a privileged user and sends them to the corresponding directory.

Note: In deployments where both directories do not use the same hashing algorithm, password synchronization is not available in an out-of-the-box installation of Oracle Internet Directory. You must configure it.

In deployments where Oracle Internet Directory is not the central directory, the password policy is enforced by the third-party directory. When there is an authentication request to the third-party directory, the latter replies that the authentication either succeeded or failed. However, any detailed password policy errors from the third-party directory are not delivered to Oracle Internet Directory and then to the client applications.

See Also: The following chapter for information about plug-ins:

- The chapter in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* about the directory plug-in framework
- The chapter in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* about customizing the external authentication plug-in

16.2.5 Choose the Structure of the Directory Information Tree

At installation, each directory server creates a default domain and a default **directory information tree (DIT)** structure. The Oracle Internet Directory infrastructure installation creates a default realm with designated containers for storing enterprise users and groups. When integrating with a connected directory, you must create identical DIT structures in both directories to use the default installation of Oracle Internet Directory. Alternatively, you can perform domain-level mapping.

This section contains these topics:

- Create Identical DIT Structures on Both Directories
- Distinguished Name Mapping and Limitations

16.2.5.1 Create Identical DIT Structures on Both Directories

Oracle recommends that you configure identical DITs on both directories. This enables all the user and group objects to be synchronized as they are, and eliminates the task of

mapping entries with distinguished names in one directory to URLs in the other. It also eliminates the performance problems that those mappings can cause.

To create identical DITs, first decide which directory is the central enterprise directory, and then change the DIT of the other one to match. Be sure to update the directory integration profile to reflect the domain-level rules.

To enable users to access Oracle applications through Oracle Application Server Single Sign-On, Oracle recommends that you identify the DIT as a separate identity management realm with its own authentication and authorization domain.

> **See Also:** The chapter about deploying identity management realms in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*

16.2.5.2 Distinguished Name Mapping and Limitations

If it is not feasible to have identical DITs on both directories, then you need to map the domains between Oracle Internet Directory and the connected directory. For example, suppose that all entries under the container `dc=mydir,dc=com` must be synchronized under `dc=myoid,dc=com` in Oracle Internet Directory. To achieve this, you specify it in the domain-level mapping rules.

If the objective is to synchronize all users and groups, then all user entries can be synchronized with the appropriate DN mapping. However, group entry synchronization can be both time consuming and carry some additional limitations. This section provides examples of both user and group synchronization when there is a DN mapping.

16.2.5.2.1 Example: User Entry Mapping Suppose that, in a mapping file, the entries in the Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server) have the format `uid=name,ou=people,o=iplanet.org`. Suppose further that the entries in Oracle Internet Directory have the format `cn=name,cn=users,dc=iplanet,dc=com`. Note that the naming attribute on Oracle Directory Server Enterprise Edition is `uid`, but on Oracle Internet Directory it is `cn`.

The mapping file has rules similar to these:

```
DomainRules
ou=people,o=iplanet.org: cn=users,dc=iplanet,dc=com: cn=%,
cn=users,dc=iplanet,dc=com
AttributeRules
Uid:1: :person:cn: :inetorgperson:
```

The value of `1` in the second column of the last line indicates that, for every change to be propagated from Oracle Directory Server Enterprise Edition to Oracle Internet Directory, the `uid` attribute must be present. This is because the `uid` must be available for constructing the DN of the entry in Oracle Internet Directory.

16.2.5.2.2 Example: Group Entry Mapping When there is a DN mapping, synchronizing group entries is somewhat complex. The group memberships, which are DNs, must have valid DN values after synchronization. This means that whatever DN mapping was done for user DNs must be applied to group membership values.

For example, suppose that the user DN values are mapped as follows:

```
ou=people,o=iplanet.org: cn=users,dc=iplanet,dc=com:
```

This implies that all the user entries under `ou=people,o=iplanet.org` are moved to `cn=users,dc=iplanet,dc=com`.

Group memberships need to be mapped as follows:

```
uniquemember: : : groupofuniquenames: uniquemember:
:groupofuniquenames:dnconvert(uniquemember)
```

For example, if the value of `uniquemember` is `cn=testuser1,ou=people,o=iplanet.org`, then it becomes `cn=testuser1,cn=users,dc=iplanet,dc=com`.

Moreover, if the value of `uniquemember` is `cn=testuser1,dc=subdomain,ou=people,o=iplanet.org`, then it becomes `cn=testuser1,dc=subdomain,cn=users,dc=iplanet,dc=com`.

This is a feasible solution as long as the naming attribute or RDN attribute remains the same on both the directories. However, if the naming attribute is different on different directories—as, for example, `ou=people,o=iplanet.org:cn=users,dc=iplanet,dc=com:cn=%,cn=users,dc=iplanet,dc=com`–then deriving the actual DNs for group memberships is not achievable through the given set of mapping rules. In this case, DN mapping for the `uniquemember` or other DN type attributes is not currently feasible.

If you want to synchronize group memberships, remember to keep the naming attribute in the source and destination directories the same.

> **See Also:** "Configuring Mapping Rules" on page 6-3 for instructions about how to specify a mapping rule

16.2.6 Select the Attribute for the Login Name

The attribute for the login name contains the identity of the end user when logging into any Oracle component. It is stored in the Oracle back-end directory as the value of the attribute `orclcommonnicknameattribute`.

If the Oracle back-end directory is Oracle Internet Directory, the attribute `orclcommonnicknameattribute` is located under the container `cn=common,cn=products,cn=oracleContext,`*`identity_management_realm`*.

If the Oracle back-end directory is Oracle Unified Directory or Oracle Directory Server Enterprise Edition, the attribute `orclcommonnicknameattribute` is located under the container `cn=common,<suffix>,`*`identity_management_realm`*.

By default, `orclcommonnicknameattribute` attribute has `uid` as its value. This means that the identity used to log in is stored in the `uid` attribute of the user entry.

If the connected directory has a specific attribute for logging in, then that attribute needs to be mapped to the right `orclcommonnicknameattribute` in Oracle Internet Directory. This needs to be one of the mapping rules in the mapping file for the connector associated with synchronizing with the connected directory.

For example, suppose that you are synchronizing Oracle Internet Directory with Microsoft Active Directory, and that, in the latter, the login identifier is contained in the `userPrincipalName` attribute of the user entry. You would synchronize the value of the `userPrincipalName` attribute to Oracle Internet Directory, storing it in the `uid` attribute, which is the value of the `orclcommonnicknameattribute` attribute. This mapping needs to be reflected in the mapping rules in the directory integration profile.

You can also use any other attribute for the login identifier. For example, if you want to use `employeeID` for logins, then mapping rules can be set accordingly. Doing this does not affect your configuration.

Note: The `orclcommonnicknameattribute` attribute is used extensively by Oracle Application Server Single Sign-On, so be sure to plan carefully how you intend to map the attribute to a connected directory attribute. After you modify this attribute, you must refresh Oracle Application Server Single Sign-On for the change to take effect.

See Also: *Oracle Fusion Middleware Guide to Delegated Administration for Oracle Identity Management* for instructions about setting the attribute for login name

16.2.7 Select the User Search Base

The user search context is represented by a multivalued attribute that lists all the containers under which users exist. Depending on your deployment, either set the user search context value to cover the entire user population, or add the container to the user search context attribute by using the Oracle Internet Directory Self-Service Console.

See Also: *Oracle Fusion Middleware Guide to Delegated Administration for Oracle Identity Management* for instructions about setting the user search context

16.2.8 Select the Group Search Base

The group search context is represented by a multivalued attribute that lists all the containers under which groups exist. Depending on your deployment, either set the group search context value to cover all group entries, or add the container to the group search context attribute by using the Oracle Internet Directory Self-Service Console.

See Also: *Oracle Fusion Middleware Guide to Delegated Administration for Oracle Identity Management* for instructions about setting the group search context

16.2.9 Decide How to Address Security Concerns

There are three main security concerns you need to consider:

- Access policies—The user and group search bases should be appropriately protected from access by any malicious users.
- Synchronization—You can configure the Oracle Directory Integration Platform to use SSL when connecting to Oracle Internet Directory and connected directories. If you do this, then all information exchanged among the directory servers is secure.
- Password synchronization—Depending on the configuration, passwords can be synchronized. For example, when Oracle Internet Directory is the central enterprise directory, password changes can be communicated to the connected directory. If passwords are to be synchronized, then Oracle recommends that you configure communication between the directories in SSL server authentication mode.

16.2.10 Administering Your Deployment with Oracle Access Manager

To use Oracle Access Manager to administer an Oracle Internet Directory deployment that synchronizes with a connected directory, you must ensure that synchronized users are visible with Oracle Access Manager.

> **See Also:** *Oracle Access Manager Identity and Common Administration Guide* for information about how to administer users in Oracle Access Manager

16.3 Microsoft Active Directory Integration Concepts

This section contains additional considerations for integrating the Oracle back-end directory with Microsoft Active Directory. It contains these topics:

- Synchronizing from Microsoft Active Directory to the Oracle Back-end Directory
- Requirement for Using WebDAV Protocol
- Windows Native Authentication
- Oracle Back-end Directory Schema Elements for Microsoft Active Directory
- Integration with Multiple Microsoft Active Directory Domain Controllers
- Synchronizing with a Multiple-Domain Microsoft Active Directory Environment
- Foreign Security Principals

> **See Also:** Chapter 18, "Integrating with Microsoft Active Directory"

16.3.1 Synchronizing from Microsoft Active Directory to the Oracle Back-end Directory

To synchronize changes from Microsoft Active Directory to the Oracle back-end directory, Oracle Directory Integration Platform imports incremental changes made available by Microsoft Active Directory change tracking mechanisms. Oracle Directory Integration Platform supports the following two Microsoft Active Directory change tracking mechanisms:

- The DirSync approach, which uses an LDAP control that is supported by Microsoft Active Directory
- The USN-Changed approach, which uses an attribute of the entry

In each approach, the directory from which changes are derived is queried at scheduled intervals by Microsoft Active Directory Connector. Each approach has advantages and disadvantages. Table 16–3 compares the two approaches.

Table 16–3 Comparing the DirSync Approach to the USN-Changed Approach

Considerations	DirSync Approach	USN-Changed Approach
Change key	Presents changes to the `ObjectGUID`, the unique identifier of the entry	Presents changes to the distinguished name. The `ObjectGUID` is used to keep track of modifications of the DN.
Error handling	If synchronization stops as a result of an error condition, then, during the next cycle, all changes that are already applied are read and skipped.	Does not require synchronization to be atomic. If synchronization stops, then the next synchronization cycle starts from the entry where the synchronization was interrupted.
Information in the search results	Changes consist of only the changed attributes and the new values. This can be quicker than the USN-Changed approach.	All attributes of the changed entry are retrieved. The retrieved values are compared to the old values stored in the Oracle back-end directory and updated. This can be more time consuming than the DirSync approach.
Changes to multivalued attributes	Reflects incremental changes made to multivalued attributes as a complete replacement of the attribute value.	Reflects incremental changes made to multivalued attributes as a complete replacement of the attribute value.
How synchronization point is tracked	When queried for changes in the directory, presents incremental changes based on a cookie value that identifies the state of the directory.	The changes are queried in the directory based on the `USNChanged` attribute, which is a long integer, that is, 8 bytes. You can modify the value to adjust where to start the synchronization.
Required user privileges	Requires the user to have the Replicate Changes privilege on the naming context of interest. This enables reading all objects and attributes in Microsoft Active Directory regardless of the access protections on them. See Also: The Microsoft Knowledge Base Article 303972 available at `http://support.microsoft.com/` for instructions on how to assign privileges to Microsoft Active Directory users when using the DirSync approach. Apply to this context the instructions used for Microsoft Active Directory management agent in this article.	Requires the Microsoft Active Directory user to have the privilege to read all required attributes to be synchronized to the Oracle back-end directory. See Also: Microsoft networking and directory documentation available in the Microsoft library at the following URL: `http://msdn.microsoft.com/` for instructions about how to assign privileges to Microsoft Active Directory users when using the USN-Changed approach.
Support of multiple domains	Requires separate connections to different domain controllers to read changes made to the entries in different domains.	Can obtain changes made to the multiple domains by connecting to the Global Catalog server. See Also: "Synchronizing with a Multiple-Domain Microsoft Active Directory Environment" on page 16-27

Table 16–3 (Cont.) Comparing the DirSync Approach to the USN-Changed Approach

Considerations	DirSync Approach	USN-Changed Approach
Synchronization from a replicated directory when switching to a different Microsoft Active Directory domain controller	Synchronization can continue. The synchronization key is the same when connecting to a replicated environment.	Requires: ■ Full synchronizing to a known point ■ Updating the `USNChanged` value ■ Starting synchronization with the failover directory See Also: "Switching to a Different Microsoft Active Directory Domain Controller in the Same Domain" on page 18-19
Synchronization scope	Reads all changes in the directory, filters out changes to the required entries, and propagates to the Oracle back-end directory.	Enables synchronization of changes in any specific subtree
Usability in an environment with multiple Microsoft Active Directory servers behind a load balancer	-	Either connect to a specific Microsoft Active Directory domain controller, or connect to a Global Catalog. Connect to Global Catalog if: ■ You are interested in import operations only ■ The Global Catalog contains all entries and attributes to be synchronized ■ Performance of the Global Catalog is acceptable

See Also: "Synchronizing from the Back-end Directory to a Connected Directory" on page 5-3

16.3.2 Requirement for Using WebDAV Protocol

If you are using the WebDAV protocol, you must configure your applications for SSL. Basic authentication is necessary because the only way for the Oracle back-end directory to authenticate the end user is to pass the plain text password to Active Directory for verification. When basic authentication is not present, digest authentication is used. But with digest authentication, the Oracle back-end directory does not have the plain text password to pass to Active Directory for verification, and therefore, end users cannot be authenticated. Basic authentication is not supported over HTTP without secure sockets layer (SSL), because the communications channel between the end user and the server would not be encrypted and the end user password would be transmitted similarly unencrypted.

16.3.3 Windows Native Authentication

This section describes how Windows Native Authentication can be used with the Oracle Directory Integration Platform. It contains these topics:

- Understanding Windows Native Authentication
- Authenticating Users Against Multiple Microsoft Active Directory Domains
- Overriding an Application Authentication Mechanism with Windows Native Authentication

16.3.3.1 Understanding Windows Native Authentication

Windows Native Authentication is an authentication scheme for users of Microsoft Internet Explorer on Microsoft Windows. When this feature is enabled in OracleAS Single Sign-On Server, users log in to OracleAS Single Sign-On Server partner applications automatically. To do this, they use Kerberos credentials obtained when the user logged in to a Windows domain.

Using the Simple and Protected GSS-API Negotiation Mechanism (SPNEGO) protocol, Internet Explorer version 5.0 and later can automatically pass the user's Kerberos credentials to a requesting Kerberos-enabled Web server. The Web server can then decode the credentials and authenticate the user.

You cannot use Microsoft integrated security or any other type of security mechanism when integrating Oracle Application Server Single Sign-On with Windows Native Authentication. Although the SPNEGO protocol supports both Kerberos version 5 and NT Lan Manager (NTLM) authentication schemes, Oracle Application Server 11*g* Release 1 (11.1.1) supports only Kerberos V5 with SPNEGO.

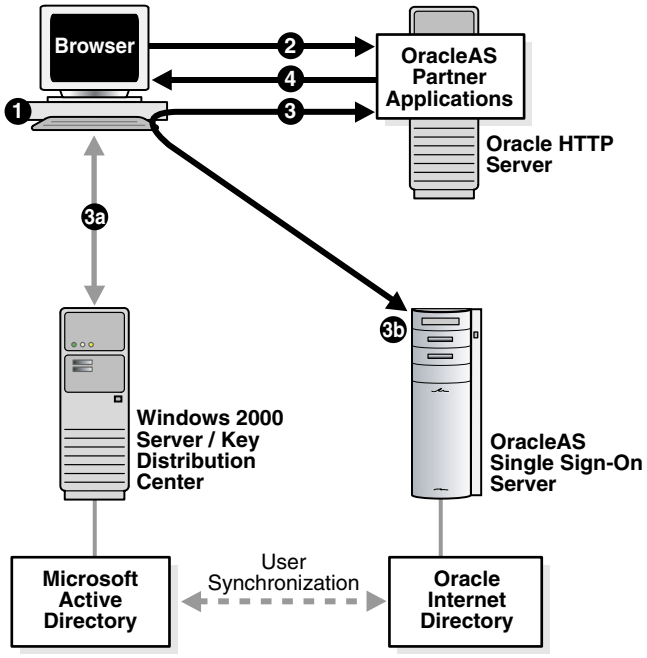
Note: Although this chapter refers only to Windows 2000, Windows Native Authentication is also supported on the Windows XP platform.

If the browser is not Internet Explorer 5.0 or higher, then Oracle Identity Management authenticates the user by using OracleAS Single Sign-On Server. Authentication to an external directory is performed by using an external authentication plug-in.

The following steps, shown in Figure 16–5 on page 16-24, describe what happens when a user tries to access a single-sign-on-protected application:

1. The user logs in to a Kerberos realm, or domain, on a Windows computer.
2. The user attempts to access a single-sign-on partner application using Internet Explorer.
3. The application routes the user to the single sign-on server for authentication. As part of this routing, the following occurs:
   a. The browser obtains a Kerberos session ticket from the Key Distribution Center (KDC).
   b. The OracleAS Single Sign-On Server verifies the Kerberos session ticket and, if the user is authorized, then the user is allowed to access the requested URL.
4. The application provides content to the user.

Figure 16–5 Flow for Windows Native Authentication



This illustration shows Oracle Application Server partner applications, the Windows 2000 server, and the OracleAS Single Sign-On Server located on separate computers. There is a box attached to the Windows server that represents Microsoft Active Directory. There is a box attached to the OracleAS Single Sign-On Server that represents the Oracle back-end directory. (The back-end directory must be Oracle Internet Directory to work with the OracleAS Single Sign-On Server.) A dotted-line arrow runs between these two boxes and it represents user synchronization between the two directories. There are standard flow arrows and an arrow showing the Kerberos ticket being granted to the browser.

****

When the user logs out of the Windows session, this application and any single sign-on applications accessed are logged out at the same time.

To use Windows Native Authentication in deployments where Microsoft Active Directory is the central directory, a user must exist in Microsoft Active Directory. If Windows Native Authentication is enabled, then, for local Oracle back-end directory users to invoke the single sign-on server, you must populate the attributes `orclsamaccountname` and `krbprincipalname` for each user entry.

16.3.3.2 Authenticating Users Against Multiple Microsoft Active Directory Domains

To authenticate users against multiple Microsoft Active Directory domains that are part of a single forest, create a global catalog and have Oracle Application Server Single Sign-On connect to the global catalog for authentication. However, if the domains are not part of the same forest, then you must create domain trusts between the domains. For detailed configuration procedures, refer to "Configuring Windows Native Authentication" on page 18-8.

16.3.3.3 Overriding an Application Authentication Mechanism with Windows Native Authentication

Windows Native Authentication does not automatically override an application's existing authentication mechanism. To use Windows Native Authentication and Oracle Application Server Single Sign-On with an application that contains an internal authentication mechanism, you must perform one of the following tasks:

- Remove the application's internal authentication mechanism.
- Configure the application as an Oracle Application Server Single Sign-On external application. This requires storing a valid application user name and password in the application configuration, making the authentication process transparent to the user after he or she logs in with Oracle Application Server Single Sign-On. For more information, refer to the *Oracle Enterprise Single Sign-On Suite Plus Administrator's Guide*.

16.3.4 Oracle Back-end Directory Schema Elements for Microsoft Active Directory

Table 16–4 lists the schema elements in the Oracle back-end directory for users that are imported from Microsoft Active Directory.

Table 16–4 Oracle Back-end Directory Schema Elements for Microsoft Active Directory

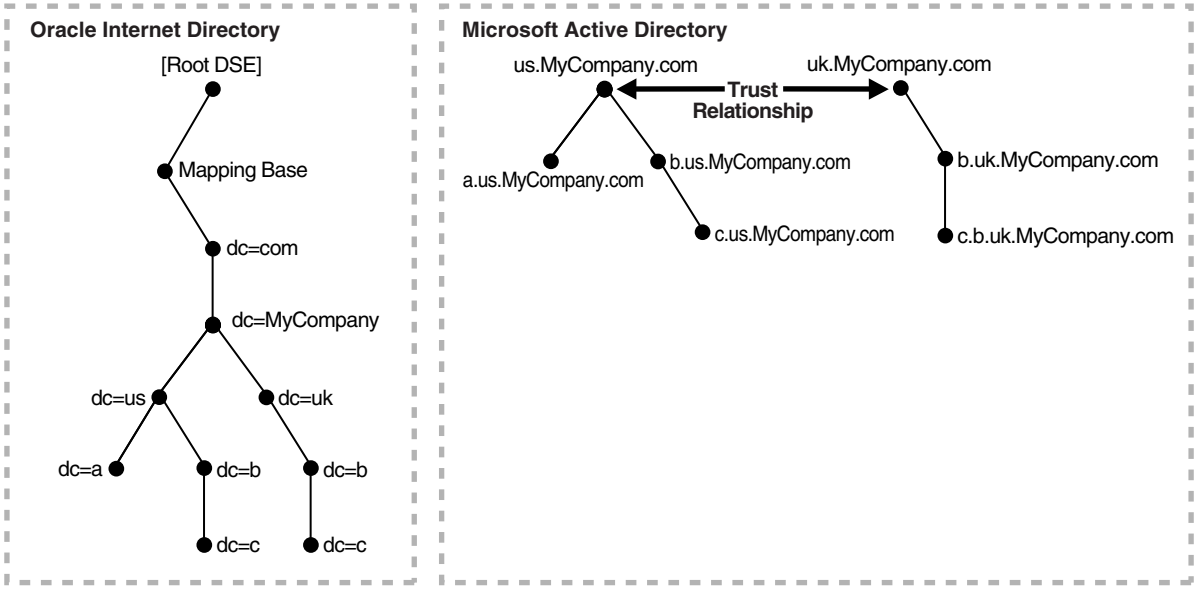
Schema Element	Description
`orclObjectGUID`	Stores Microsoft Active Directory's `OBJECTGUID` attribute value for users and groups migrated to the Oracle back-end directory from Microsoft Active Directory.
`orclObjectSID`	Stores Microsoft Active Directory's `OBJECTSID` attribute value for users and groups migrated to the Oracle back-end directory from Microsoft Active Directory.
`orclSAMAccountName`	Stores the value of Microsoft Active Directory's `SAMAccountName` attribute. In the Oracle back-end directory, this attribute is defined as a directory string type. However, in Microsoft Active Directory this attribute cannot accept any special or non-printable characters. If any entry is added in the Oracle back-end directory with this attribute, it can only contain a simple text string or synchronization from the Oracle back-end directory to Microsoft Active Directory will fail.
`orclUserPrincipalName`	Stores the Kerberos user principal name for Microsoft Active Directory users.
`orclADGroup`	Contains Microsoft Active Directory group attributes, which are used to synchronize Microsoft Active Directory group objects with the Oracle back-end directory group objects in an Oracle Directory Integration environment.
`orclADUser`	Contains Microsoft Active Directory user attributes, which are used to synchronize Microsoft Active Directory user objects with the Oracle back-end directory user objects in an Oracle Directory Integration and Provisioning environment.
`orclSourceObjectDN`	Represents the DN for the respective entry in Microsoft Active Directory. This value is required to perform external authentication if different domains are mapped between both directories.

See Also: *Oracle Fusion Middleware Reference for Oracle Identity Management* for detailed information about the Oracle Internet Directory schema elements for Microsoft Active Directory

16.3.5 Integration with Multiple Microsoft Active Directory Domain Controllers

A deployment of Microsoft Active Directory with multiple domains can have either a single DIT or a combination of two or more DITs. In Microsoft Active Directory, a group of DITs is called a forest. Figure 16–6 shows how a forest in Microsoft Active Directory is reflected in the Oracle back-end directory.

Figure 16–6 Mapping Between the Oracle Back-end Directory and a Forest in Microsoft Active Directory



This illustration shows a directory information tree (DIT) in the Oracle back-end directory and a corresponding forest in Microsoft Active Directory. The DIT in the Oracle back-end directory extends downward from the root DSE to a node labeled `Mapping Base` to `dc=com` to `dc=MyCompany`. From there it divides into two branches: `dc=us` and `dc=uk`. The node `dc=us` extends downward in two branches: `dc=a` and `dc=b`. The `dc=b` node extends downward to `dc=c`. The node `dc=b` extends to `dc=c`. The node `dc=uk` extends downward to `dc=b` and then to `dc=c`. The forest in Microsoft Active Directory has two roots: `us.MyCompany.com` and `uk.MyCompany.com`. The node `us.MyCompany.com` divides into two branches: `a.us.MyCompany.com` and `b.us.My.Company.com`. The latter node extends downward to `c.us.MyCompany.com`. The node `uk.MyCompany.com` extends downward to `b.uk.MyCompany.com` and from there to `c.uk.MyCompany.com`.

****

In this directory, two domain trees constitute a forest. These trees are in a trust relationship, that is, users in one domain are authenticated by the domain controller in the other domain. This forest in Microsoft Active Directory maps to an identically structured subtree in the Oracle back-end directory.

Considerations for Deployments where the Oracle Back-end Directory is the Central Directory

If there are multiple Microsoft Active Directory domains, the `syncProfileBootstrap` command must be run as many times as there are Microsoft Active Directory domains. Each time you do this, you choose the specific data set required by the target Microsoft Active Directory domain.

The Oracle Directory Integration Platform provisions users and groups in the respective Microsoft Active Directory domains. Before provisioning can take place, you must configure a one-way synchronization from the Oracle back-end directory to the Microsoft Active Directory domain.

Considerations for Deployments where Microsoft Active Directory as the Central Directory

If there are multiple Microsoft Active Directory servers, then you must bootstrap the data from each Microsoft Active Directory domain. If you use the Global Catalog for one-way synchronization from Microsoft Active Directory to the Oracle back-end directory, then you need to bootstrap only once from the Global Catalog server.

The Oracle Directory Integration Platform synchronizes users and groups from the respective Microsoft Active Directory domains into the Oracle back-end directory. Before the provisioning can take place, a one-way synchronization between the Oracle back-end directory and a domain controller on each Microsoft Active Directory domain must be established.

16.3.6 Synchronizing with a Multiple-Domain Microsoft Active Directory Environment

This section describes considerations for synchronizing with a multiple-domain Microsoft Active Directory environment. It contains these topics:

- Configuration Required for Importing from Microsoft Active Directory to the Oracle Back-end Directory
- Configuration Required for Importing from Microsoft Active Directory Lightweight Directory Service to the Oracle Back-end Directory
- Configuration Required for Exporting from the Oracle Back-end Directory to Microsoft Active Directory
- Example: Integration with Multiple Connected Directory Domains

16.3.6.1 Configuration Required for Importing from Microsoft Active Directory to the Oracle Back-end Directory

Normally, importing requires configuring one import profile for each Microsoft Active Directory domain regardless of whether you are using the DirSync approach or the USN-Changed approach. However, if you are using the USN-Changed approach, you can use the Global Catalog to import from an entire Microsoft Active Directory forest. You only need to configure a single import profile to use Global Catalog, but keep in mind the following considerations:

- Because Global Catalog is read-only, you can use it only for importing data into the Oracle back-end directory
- Global Catalog does not contain all the attributes, although the available attributes can be configured in Microsoft Active Directory
- Because Global Catalog is a point of authentication, you may incur additional overhead if synchronization is started from this point

> **See Also:** The Microsoft Knowledge Base Article 256938 available from Microsoft Help and Support at `http://support.microsoft.com/` for information about Global Catalog attributes in the Microsoft Active Directory schema

16.3.6.2 Configuration Required for Importing from Microsoft Active Directory Lightweight Directory Service to the Oracle Back-end Directory

Unlike Microsoft Active Directory, only the USN changed approach is used for synchronizing from Microsoft Active Directory Lightweight Directory Service (AD LDS), which was previously known as Active Directory Application Mode or ADAM, to the Oracle back-end directory. To import entries from Microsoft AD LDS to the Oracle back-end directory, you must configure an import profile connecting to Microsoft AD LDS with the respective port details.

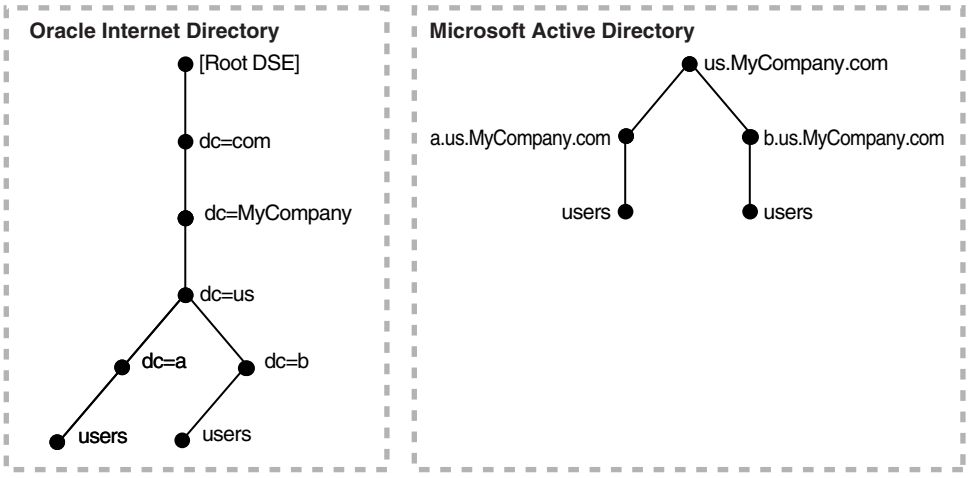
16.3.6.3 Configuration Required for Exporting from the Oracle Back-end Directory to Microsoft Active Directory

To integrate with multiple-domain Microsoft Active Directory environments, the Oracle Directory Integration Platform obtains configuration information from each Microsoft Active Directory domain. You must configure as many export profiles as there are Microsoft Active Directory domains.

16.3.6.4 Example: Integration with Multiple Connected Directory Domains

A deployment of a connected directory with multiple domains can have either a single DIT or a combination of two or more DITs. Figure 16–7 shows how multiple domains in a connected directory are mapped to a DIT in the Oracle back-end directory.

Figure 16–7 Example of a Mapping Between the Oracle Back-end Directory and Multiple Domains in Microsoft Active Directory



This illustration shows a directory information tree (DIT) in the Oracle back-end directory and corresponding multiple domains in Microsoft Active Directory. The DIT in the Oracle back-end directory extends downward from the root DSE to `dc=com` to `dc=MyCompany` to `dc=us`. From there it divides into two branches: `dc=a` and `dc=b`. The node `dc=a` extends to `users`, and, similarly, the node `dc=b` extends to `users`. The multiple domains in Microsoft Active Directory has `us.MyCompany.com` as the root. From there it extends downward to `a.us.MyCompany.com` and `b.us.MyCompany.com`. Each of the latter two nodes extends downward to a node labeled `users`.

****

In Figure 16–7, the connected directory environment has a parent and two children.

The first child domain `a.us.MyCompany.com` maps to `dc=a,dc=us,dc=MyCompany,dc=com` in the Oracle back-end directory. The second child domain `b.us.MyCompany.com` maps to `dc=b,dc=us,dc=MyCompany,dc=com` in the Oracle back-end directory. The common domain component in the connected directory environment `us.MyCompany.com` maps to the default identity management realm in the Oracle back-end directory, in this case `dc=us,MyCompany,dc=com`.

16.3.7 Foreign Security Principals

A Microsoft Active Directory user or computer account represents a physical entity such as a computer or person. User accounts and computer accounts, as well as groups, are called security principals. Security principals are directory objects that are automatically assigned security identifiers. Objects with security identifiers can log on to the network and access domain resources. A user or computer account is used to:

- Authenticate the identity of the user or computer
- Authorize or deny access to domain resources
- Administer other security principals
- Audit actions performed using the user or computer account

For example, the user and computer accounts that are members of the Enterprise Administrators group are automatically granted permission to log on at all of the domain controllers in the forest.

User and computer accounts are added, disabled, reset, and deleted by using Microsoft Active Directory Users and Computers.

In a trust relationship in Microsoft Active Directory, users in one domain are authenticated by a domain controller in another domain. The trust relationship can be transitive or non transitive.

- In a transitive trust relationship, the trust relationship extended to one domain is automatically extended to all other domains that trust that domain. For example, suppose you have three domains: A, B, and C in which both B and C are in a direct trust relationship with A. In this scenario, both B and C also trust each other. This is because, although they are not in a direct trust relationship with each other, they are in a direct trust relationship with A.
- In a non transitive trust relationship, the trust is bound by the two domains in the trust relationship; it does not flow to any other domains in the forest.

When a trust is established between a Windows 2000 domain in a particular forest and a Windows 2000 domain outside of that forest, security principals from the external domain can be granted access to resources in the forest. A security principal from an external domain is called a *foreign security principal* and is represented in Microsoft Active Directory as a "foreign security principal" object. These foreign security principals can become members of domain local groups, which can have members from domains outside of the forest.

Foreign security principals are used when there is a non transitive trust between two domains in a Microsoft Active Directory environment.

In a non-transitive trust relationship in a Microsoft Active Directory environment, when one domain recognizes a foreign security principal from the other domain, it represents that entity similar to a DN entry. In that entry, the RDN component is set to the SID of the original entry in the trusted domain. In the case of groups, the DNs of the foreign security principals are represented as member values, not as the DNs of the

original entries in the trusted domain. This can create a problem when foreign security principals are synchronized with the Oracle back-end directory.

16.4 Oracle Directory Server Enterprise Edition (Sun Java System Directory Server) Integration Concepts

This section contains additional considerations for integrating Oracle Internet Directory with Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server). It contains these topics:

- Synchronizing from Oracle Directory Server Enterprise Edition to Oracle Directory Integration Platform
- Oracle Internet Directory Schema Elements for Oracle Directory Server Enterprise Edition (Sun Java System Directory Server)

> **See Also:** Chapter 20, "Integrating with Oracle Directory Server Enterprise Edition (Sun Java System Directory Server)"

16.4.1 Synchronizing from Oracle Directory Server Enterprise Edition to Oracle Directory Integration Platform

Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server) maintains a change log in which it stores incremental changes made to directory objects. Synchronization from Oracle Directory Server Enterprise Edition to Oracle Internet Directory makes use of this change log.

> **See Also:**
>
> - "Synchronizing from the Back-end Directory to a Connected Directory" on page 5-3.
> - The Oracle Internet Directory server administration tools chapter of the *Oracle Identity Management User Reference* for instructions on how to start an Oracle Internet Directory server with change logging enabled.
> - Oracle Directory Server Enterprise Edition documentation for instructions about how to configure change logging. If you plan to synchronize with either Sun Java System Directory Server versions 5.0 or later, or Oracle Directory Server Enterprise Edition, the retro change log plug-in must be enabled.

16.4.2 Oracle Internet Directory Schema Elements for Oracle Directory Server Enterprise Edition (Sun Java System Directory Server)

Oracle Internet Directory includes the `orclSourceObjectDN` attribute for users that are imported from Oracle Directory Server Enterprise Edition. The `orclSourceObjectDN` element represents the DN for the respective entry in Oracle Directory Server Enterprise Edition. This value is required to perform external authentication if different domains are mapped between both directories.

16.5 IBM Tivoli Directory Server Integration Concepts

This section contains additional considerations for integrating the Oracle back-end directory with IBM Tivoli Directory Server. It contains these topics:

- Changes to Directory Objects in IBM Tivoli Directory Server
- Oracle Back-end Directory Schema Elements for IBM Tivoli Directory Server

16.5.1 Changes to Directory Objects in IBM Tivoli Directory Server

IBM Tivoli Directory Server maintains a change log where it stores incremental changes made to directory objects. Synchronization from IBM Tivoli Directory Server to the Oracle back-end directory makes use of this change log.

Note: Tombstone is supported in IBM Tivoli Directory Server version 6.2.

16.5.2 Oracle Back-end Directory Schema Elements for IBM Tivoli Directory Server

Table 16–5 lists the schema elements in the Oracle back-end directory for users that are imported from IBM Tivoli Directory Server:

Table 16–5 Oracle Back-end Directory Schema Elements for IBM Tivoli Directory Server

Schema Element	Description
`orclSourceObjectDN`	Represents the DN for the respective entry in Tivoli. This value is required to perform external authentication if different domains are mapped between both directories.
`orclTDSEntryUUID`	Represents the entryUUID value for the respective entry in IBM Tivoli. This value is used as the synchronization key.
`orclTDSObject`	Represents the Tivoli directory object.

See Also:

- "Synchronizing from the Back-end Directory to a Connected Directory" on page 5-3.
- The Oracle Internet Directory server administration tools chapter of the *Oracle Identity Management User Reference* for instructions on how to start an Oracle Internet Directory server with change logging enabled.
- IBM Tivoli Directory Server documentation for instructions on how to configure change logging.

16.6 Novell eDirectory and OpenLDAP Integration Concepts

This section contains additional considerations for integrating the Oracle back-end directory with Novell eDirectory or OpenLDAP. It contains these topics:

- Synchronizing from Novell eDirectory or OpenLDAP to the Oracle Back-end Directory
- Oracle Back-end Directory Schema Elements for Novell eDirectory
- Oracle Back-end Directory Schema Elements for OpenLDAP

See Also: Chapter 22, "Integrating with Novell eDirectory or OpenLDAP"

16.6.1 Synchronizing from Novell eDirectory or OpenLDAP to the Oracle Back-end Directory

To synchronize changes from Novell eDirectory or OpenLDAP to the Oracle back-end directory, the Oracle Directory Integration Platform evaluates the modification timestamp of each Novell eDirectory or OpenLDAP entry. Entries with timestamps that are more recent than the execution time of the last synchronization are updated in the Oracle back-end directory.

For entries that have been deleted in Novell eDirectory or OpenLDAP, the Oracle Directory Integration Platform identifies the deleted entries by performing a linear comparison between the entries in the Oracle back-end directory and Novell eDirectory or OpenLDAP. In other words, entries in both directories are compared at specified intervals. Entries that are not available in both the Oracle back-end directory and Novell eDirectory or OpenLDAP are deleted. To avoid decreased performance on the server as directory entries are compared, you can customize the comparison to search specific subsets of the DIT.

See Also:

- "Synchronizing from the Back-end Directory to a Connected Directory" on page 5-3
- "Step 6: Customize the Novell eDirectory or OpenLDAP Connector to Synchronize Deletions" on page 22-5 for information about how to search specific subsets of the DIT when synchronizing deletions between the Oracle back-end directory and Novell eDirectory or OpenLDAP

16.6.2 Oracle Back-end Directory Schema Elements for Novell eDirectory

Table 16–6 lists the schema elements in the Oracle back-end directory for users that are imported from Novell eDirectory.

Table 16–6 Oracle Back-end Directory Schema Elements for Novell eDirectory

Schema Element	Description
`orclSourceObjectDN`	Represents the DN for the respective entry in Novell eDirectory. This value is required to perform external authentication if different domains are mapped between both directories.
`orclndsobjectguid`	Required for reconciliation. Represents the GUID value for the respective entry in Novell eDirectory. This value is used as the synchronization key.
`orclsourcemodifytimestamp`	Required. Represents the `modifytimestamp` attribute of the respective entry in Novell eDirectory. This value is used in getting the entries that needs to be synchronized.
`orclsourceCreateTimestamp`	Required. Represents the `createtimestamp` attribute of the respective entry in Novell eDirectory. This value is used in synchronization of deleted entries.
`orclndsobject`	Represents the NDS object in Novell eDirectory.

See Also: *Oracle Fusion Middleware Reference for Oracle Identity Management* for detailed information about the Oracle back-end directory schema elements for Novell eDirectory.

16.6.3 Oracle Back-end Directory Schema Elements for OpenLDAP

Table 16–7 lists the schema elements in the Oracle back-end directory for users that are imported from OpenLDAP.

Table 16–7 Oracle Internet Directory Schema Elements for OpenLDAP

Schema Element	Description
`orclSourceObjectDN`	Represents the DN for the respective entry in OpenLDAP. This value is required to perform external authentication if different domains are mapped between both directories.
`orclOpenLdapEntryUUID`	Required for reconciliation. Represents the entryUUID value for the respective entry in OpenLDAP. This value is used as the synchronization key.
`orclsourcemodifytimestamp`	Required. Represents the `modifytimestamp` attribute of the respective entry in OpenLDAP. This value is used in getting the entries that needs to be synchronized.
`orclsourceCreateTimestamp`	Required. Represents the `createtimestamp` attribute of the respective entry in OpenLDAP. This value is used in synchronization of deleted entries.
`orclopenldapobject`	Represents the OpenLDAP object.

> **See Also:** *Oracle Fusion Middleware Reference for Oracle Identity Management* for detailed information about the Oracle back-end directory schema elements for OpenLDAP

16.7 Limitations of Connected Directory Integration in Oracle Directory Integration Platform 11*g* Release 1 (11.1.1)

Oracle Directory Integration Platform 11*g* Release 1 (11.1.1) does not support the synchronization of the schema and ACLs. You can use the `schemasync` tool to identify differences in schema, specifically attributes and object classes, between Oracle Internet Directory and connected directories. After identifying the differences, you can use the schemasync tool to synchronize the schema.

> **See Also:** The *Oracle Fusion Middleware Reference for Oracle Identity Management* for more information about the `schemasync` tool.

17

Configuring Synchronization with a Connected Directory

This chapter contains generic instructions for synchronizing the Oracle back-end directory with a connected directory. It contains these topics:

- Verifying Synchronization Requirements
- Creating Import and Export Synchronization Profiles Using expressSyncSetup
- Configuring Advanced Integration Options
- Writing Custom Synchronization Connectors

Note: This chapter assumes that you are familiar with Chapter 16, "Connected Directory Integration Concepts and Considerations".

See Also: The following chapters for step-by-step instructions about configuring integration between the Oracle back-end directory and the following connected directories:

- Chapter 18, "Integrating with Microsoft Active Directory"
- Chapter 20, "Integrating with Oracle Directory Server Enterprise Edition (Sun Java System Directory Server)"
- Chapter 21, "Integrating with IBM Tivoli Directory Server"
- Chapter 22, "Integrating with Novell eDirectory or OpenLDAP"

17.1 Verifying Synchronization Requirements

To prepare for synchronization between the Oracle back-end directory and a connected directory, do the following:

1. Verify that the Oracle back-end directory and the other directory are running.
2. Create a user account in the connected directory with sufficient privileges to read and write the relevant entries in the containers that will be synchronized. If the directory supports tombstone, the account should also have sufficient privileges to read tombstone entries.
   - **For Import Operations from a Connected Directory**: Grant the user account read access privileges to the subtree root. The user account must be able to read all objects under the source container (subtree root) in the connected directory that are to be synchronized with the Oracle Directory Integration Platform. To verify whether a connected directory user account has the

necessary privileges to all objects to be synchronized with the Oracle back-end directory, use the command-line `ldapsearch` utility to perform a subtree search, as follows:

```
$ORACLE_HOME/bin/ldapsearch -h directory host-p directory port \
-b "DN of subtree" -s sub -D binddn "objectclass=*" -q
```

Note: You will be prompted for the password for the privileged directory user.

The return results from the `ldapsearch` utility should include all objects of interest, including all attributes and values that will be synchronized.

- **For Export Operations to a Connected Directory**: Grant the user account the following privileges to the subtree root that is the parent of all the containers to which the Oracle Directory Integration Platform will export users:
  - Write
  - Create all child objects
  - Delete all child objects

  See Also: The connected directory documentation for information about how to grant privileges to user accounts

  You must also ensure that the Oracle back-end directory is running with change logging enabled, and that the change log purge duration is set to a minimum of seven days.

  See Also:

  - The Oracle Internet Directory server administration tools chapter of the *Oracle Identity Management User Reference* for instructions about how to start an Oracle directory server with change logging enabled
  - The `orclPurgeTargetAge` section of the *Oracle Identity Management User Reference* for instructions about how to set the change log purge duration

17.2 Creating Import and Export Synchronization Profiles Using expressSyncSetup

The `expressSyncSetup` command located in the *ORACLE_HOME*`/bin` directory allows you to perform the initial migration of data between a connected directory and the Oracle back-end directory for a synchronization profile.

Notes:

- Best security practice is to provide a password only in response to a prompt from the command.
- You must set the `WLS_HOME` and `ORACLE_HOME` environment variables before executing any of the Oracle Directory Integration Platform commands.
- The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. Refer to the *Configuring SSL* chapter in *Oracle Fusion Middleware Securing Oracle WebLogic Server* for more information.

17.2.1 Syntax for expressSyncSetup

expressSyncSetup

```
expressSyncSetup -h HOST -p PORT -D wlsuser -pf PROFILE
-conDirType CONNECTED_DIRECTORY_TYPE -conDirURL CONNECTED_DIRECTORY_URL
-conDirBindDN CONNECTED_DIRECTORY_BIND_DN -conDircontainer SYNC_CONTAINER
[-ssl -keystorePath PATH_TO_KEYSTORE -keystoreType TYPE] [-enableProfiles {true |
false}] [-help]
```

17.2.2 Arguments for expressSyncSetup

-h | -host

Oracle WebLogic Server host where Oracle Directory Integration Platform is deployed.

-p | -port

Listening port of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed.

-D | wlsusser

Oracle WebLogic Server login ID

Note: You will be prompted for the Oracle WebLogic Server login password. You cannot provide the password as a command-line argument. Best security practice is to provide a password only in response to a prompt from the command. If you must execute `expressSyncSetup` from a script, you can redirect input from a file containing the Oracle WebLogic Server login password. Use file permissions to protect the file and delete it when it is no longer necessary. If you must provide more than one password to `expressSyncSetup`, put each on a separate line in the file, in the following order: connected directory bind DN password, then Oracle WebLogic Server login password.

-pf | -profile

Profile name. Specify the name of the profile in ASCII characters only, as non-ASCII characters are not supported in the profile name.

-conDirType

Connected directory type. The supported values are `ActiveDirectory`, `EDirectory`, `iPlanet`, `OpenLDAP`, `ADAM`, `Tivoli`, `OID`, and `ExchangeServer2003`.

-conDirUrl

URL where the connected directory is running. The format is *host*:*port*.

-conDirBindDN

Connected directory server bind DN. For example:

```
administrator@idm2003.net
```

```
cn=orcladmin,cn=Directory Manager
```

> **Note:** You will be prompted for the connected directory bind DN password. You cannot provide the password as a command-line argument. Best security practice is to provide a password only in response to a prompt from the command. If you must execute `expressSyncSetup` from a script, you can redirect input from a file containing the connected directory bind DN password. Use file permissions to protect the file and delete it when it is no longer necessary. If you must provide more than one password to `expressSyncSetup`, put each on a separate line in the file, in the following order: connected directory bind DN password, then Oracle WebLogic Server login password.

-conDirContainer

The synchronization container. For example:

```
ou=sales,dc=us,dc=com
```

```
OU=Groups,DC=imtest,DC=com
```

```
CN=Users,DC=imtest,DC=com
```

-ssl

Executes the command in SSL mode.

> **Note:** The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. Refer to the *Configuring SSL* chapter in *Oracle Fusion Middleware Securing Oracle WebLogic Server* for more information.

-keystorePath

The full path to the keystore.

-keystoreType

The type of the keystore identified by `-keystorePath`. For example: `-keystorePath jks` or `-keystorePath PKCS12`

-enableProfiles

Specify `true` to enable created profiles, `false` if not.

-help

Provides command usage help.

17.2.3 Tasks and Examples for expressSyncSetup

```
expressSyncSetup -h myhost.mycompany.com -p 7005 -D login_ID -pf myProfile \
  -conDirType ACTIVEDIRECTORY -conDirUrl server.mycompany.com:5432 \
  -conDirBindDN administrator@idm2003.net -conDirContainer ou=sales,dc=us,dc=com \
  -enableProfiles false \

expressSyncSetup -help
```

17.2.4 Understanding the expressSyncSetup Command

The `expressSyncSetup` command allows you to create two synchronization profiles, one for import and one for export, using predefined assumptions. If the Oracle Directory Integration Platform is already running, then after enabling the profile, you can immediately begin synchronizing users and groups between the containers in which users and groups are stored in the connected directory and the container in the Oracle back-end directory.

> **Note:** While customizing the synchronization profiles for your environment, you may need to add test users and groups to facilitate your deployment effort. Be sure to remove any test users and groups when you are finished customizing and testing your synchronization profiles.

To simplify the configuration, the `expressSyncSetup` command assumes the following:

- Entries for users of the default realm in Oracle Internet Directory are located in the container `cn=users,`*`default_realm_DN`*, whereas users of the default realm in Oracle Unified Directory and Oracle Directory Server Enterprise Edition are located in the container `cn=users,`*`default_metadata_suffix`*.
- Entries for groups of the default realm are located in the container `cn=groups,`*`default_realm_DN`* for Oracle Internet Directory, and in the container default_metadata_suffix for Oracle Unified Directory and Oracle Directory Server Enterprise Edition.
- The Oracle Directory Integration Platform master mapping rules files created during installation are located in $*ORACLE_HOME*`/ldap/odi/conf`.
- Master domain mapping rules are located in the $*ORACLE_HOME*`/ldap/odi/conf/` directory.
- The logon credential is that of an Oracle Directory Integration Platform administrator with sufficient privileges to configure a profile, a realm, and access controls on the Users container in the Oracle directory server. Members of the dipadmingrp have the necessary privileges.

In Oracle Internet Directory, the Oracle Directory Integration Platform Administrators group is as follows:

```
cn=dipadmingrp,cn=dipadmins,cn=directory integration
platform,cn=products,cn=oraclecontext
```

In Oracle Unified Directory and Oracle Directory Server Enterprise Edition, the Oracle Directory Integration Platform Administrators group is as follows:

```
cn=dipadmingrp,cn=dipadmins,cn=directory integration
platform,<suffix>
```

Perform the following steps to run the `expressSyncSetup` command and verify that users and groups are synchronizing between `cn=users,default_naming_context` in the connected directory, and `cn=users,`*`default_realm`* in the Oracle back-end directory:

1. Run express configuration using "Syntax for expressSyncSetup" on page 17-3.
2. The `expressSyncSetup` command creates two profiles named *`profile_name`*`Import` and *`profile_name`*`Export`. By default, both profiles are disabled. Enable the *`profile_name`*`Import` profile if you need to synchronize from a connected directory to the Oracle back-end directory and enable the *`profile_name`*`Export` profile if you need to synchronize from the Oracle back-end directory to a connected directory. Enable the profile by using the `manageSyncProfiles` command with the `activate` operation.
3. Wait until the scheduling interval has elapsed and verify that synchronization has started by entering the following command. After executing the command, you will be prompted for the password for privileged directory user.

```
$ORACLE_HOME/bin/ldapsearch -h OID host -p OID port \
-D binddn -q \
-b "orclodipagentname=import profile,cn=subscriber profile,cn=changelog
subscriber,cn=oracle internet directory" -s base "objectclass=*"
orclodipsynchronizationstatus orclodiplastsuccessfulexecutiontime
```

> **Note:** The default scheduling interval is 60 seconds (1 minute). You can use Oracle Enterprise Manager Fusion Middleware Control to change the default scheduling interval. See Chapter 7, "Managing Directory Synchronization Profiles" for information about using Oracle Enterprise Manager Fusion Middleware Control.

When synchronization is successfully started:

- The value of the `Synchronization Status` attribute is `Synchronization Successful`.
- The value of the `Last Successful Execution Time` attribute is the specific date and time of that execution. Note that this must be close to the current date and time.

An example of a result indicating successful synchronization is:

```
Synchronization successful 20060515012615
```

Note:

- The date and time must be close to current date and time
- When running the `ldapsearch` command, you need the dipadmin password, which, as established at installation, is the same as orcladmin password

4. After verifying that synchronization has started, examine the entries in the Oracle back-end directory and the connected directory to confirm that users and groups are synchronizing between `cn=users,default_naming_context` in the connected directory, and `cn=users,`*`default_realm`* in the Oracle back-end directory.

CAUTION: In order to successfully customize your import and export synchronization profiles, do not enable SSL until you have finished with all other configuration tasks.

17.3 Configuring Advanced Integration Options

When you install Oracle Directory Integration Platform, sample import and export synchronization profiles are automatically created for each of the supported Oracle and third-party connected directories. The import and export synchronization profiles created during the install process or with the `expressSyncSetup` command are only intended as a starting point for you to use when deploying your integration of the Oracle back-end directory and a connected directory. Because the default synchronization profiles are created using predefined assumptions, you must further customize them for your environment, as described in these topics:

- Configuring the Realm
- Customizing Access Control Lists
- Customizing Mapping Rules
- Configuring the Connected Directory Connector for Synchronization in SSL Mode
- Enabling Password Synchronization from the Oracle Back-end Directory to a Connected Directory
- Configuring External Authentication Plug-ins

See Also: The individual connected directory integration chapters for information on the sample synchronization profiles that were created during the installation process

Before customizing the sample synchronization profiles that were created during the installation process, be sure to copy them with the `copy` operation of the `manageSyncProfiles` command, then enable the copies with the `activate` operation of the `manageSyncProfiles` command.

17.3.1 Configuring the Realm

> **Note:** If your Oracle back-end directory is either Oracle Unified Directory or Oracle Directory Server Enterprise Edition, the default container in those directories is the metadata suffix. Consequently, you may need to add `cn=users, <metadata_suffix>` and `cn=groups,<metadata_suffix>` entries and update the domain mapping rules as needed.

To configure the realm, do the following:

1. Choose the realm DN structure as described in the section "Choose the Structure of the Directory Information Tree" on page 16-16, and, more specifically, in the section "Planning the Deployment" on page 16-7.
2. Select the attribute for the login name of the user. This attribute contains the name of the attribute used for logging in. By default, it is `uid`. For more information, see the section "Select the Attribute for the Login Name" on page 16-18.
   - If you are integrating with Microsoft Active Directory, and the `userprincipalname` attribute is used for logging in, then you would map `userprincipalname` to the `uid` attribute in the Oracle back-end directory.
   - If you are integrating with Novell eDirectory or OpenLDAP, and the `mail` attribute is used for logging in, then you would map `mail` to the `uid` attribute in the Oracle back-end directory.
3. Set up the `usersearchbase` and `groupsearchbase` values in the Oracle back-end directory. These values indicate to the various Oracle components where to look for users and groups in the Oracle back-end directory. They are set to default values during installation. However, you may need to reset these values so that they correspond to the DIT structures in the two directories. Be sure to set them correctly. Otherwise, even if the synchronization seems to function properly, components still may be unable to access users and groups in the Oracle back-end directory.

   To illustrate how you might configure the user search base and group search base: In the example in Figure 16–2 on page 16-8, the value of `usersearchbase` should be set to `cn=users,dc=us,dc=MyCompany,dc=com` or one of its parents. Similarly, assuming there is a subtree named `groups` in the DIT, the multivalued `groupsearchbase` attribute should be set to both of the following:

   - `cn=groups,dc=us,dc=MyCompany,dc=com` or one of its parents
   - `cn=users,dc=us,dc=MyCompany,dc=com`

   To configure the user search base and group search base, use the Oracle Internet Directory Self-Service Console.
4. Set up the `usercreatebase` and `groupcreatebase` values in the Oracle back-end directory. These values indicate to the various Oracle components where users and groups can be created. They are set to default values during installation.

   To illustrate how to configure the user create base and group create base: In the example in Figure 16–2 on page 16-8, the value of `usercreatebase` should be set to `cn=users,dc=us,dc=MyCompany,dc=com` or one of its parents. Similarly, the `groupcreatebase` should be set to `cn=groups,dc=us, dc=MyCompany,dc=com` or one of its parents.

To configure the user create base and group create base, use the Oracle Internet Directory Self-Service Console.

> **See Also:** The section about modifying configuration settings for an identity management realm in *Oracle Fusion Middleware Guide to Delegated Administration for Oracle Identity Management*

17.3.2 Customizing Access Control Lists

This section discusses how to customize ACLs for import profiles, export profiles, and for other Oracle components. It contains these topics:

- Customizing ACLs for Import Profiles
- Customizing ACLs for Export Profiles
- ACLs for Other Oracle Components

17.3.2.1 Customizing ACLs for Import Profiles

The import profile is the identity used by the Oracle Directory Integration Platform to access the Oracle back-end directory. ACLs must enable the import profile to add, modify, and delete objects in either the users and groups containers or the subtree where entries are accessed. By default, import profiles are part of the Realm Administrators group (`cn=RealmAdministrators, cn=groups,cn=OracleContext,`*`realm_DN`*) in the default realm. This group has privileges to perform all operations on any entry under the DN of the default realm.

Customizing an ACL for Oracle Unified Directory or Oracle Directory Server Enterprise Edition Back-end Directories

If your Oracle back-end directory is either Oracle Unified Directory or Oracle Directory Server Enterprise Edition, the import profile can add, modify, and delete users and groups under the DN of the metadata suffix. For a non-metadata suffix, the ACL has to be set as follows so that the containers can import the users and groups from the other source:

```
dn: <Container DN>
changetype:modify
add: aci
aci: (target="ldap:///<Container DN>")(version 3.0; acl "Anonymous read-search
access"; allow (read,add,delete,search,write,compare,proxy)
groupdn="ldap:///cn=dipadmingrp,cn=DIPadmins,cn=Directory Integration
Platform,<metadata suffix>"; allow (read,add,delete,search,write,compare,proxy)
groupdn="ldap:///cn=odipigroup,cn=DIPadmins,cn=Directory Integration
Platform,<metadata suffix>"; )
-
add: aci
aci: (targetattr="*")(version 3.0; acl "Anonymous read-search access";
allow (search,read,write,compare,add)
groupdn="ldap:///cn=dipadmingrp,cn=DIPadmins,cn=Directory Integration
Platform,<metadata suffix>"; allow (search,read,write,compare,add)
groupdn="ldap:///cn=odipigroup,cn=DIPadmins,cn=Directory Integration
Platform,<metadata suffix>";
```

Customizing an LDIF ACL for Oracle Unified Directory or Oracle Directory Server Enterprise Edition Back-end Directories

Refer to the ACL example given above. As needed, replace `<Container DN>` with the DN under which the operations are to be performed, and replace `<metadata`

suffix> with the suffix that you specified to store DIP metadata during DIP configuration.

You can upload an LDIF file using the following ldapmodify command:

```
$ORACLE_HOME/bin/ldapmodify -h OID host -p OID port
-D binddn -q -v -f realmacl.ldif
```

After executing the command, you will be prompted for the password for privileged directory user.

Customizing an ACL for an Oracle Internet Directory Back-end Directory

If your Oracle back-end directory is Oracle Internet Directory, you should not need to customize the ACLs for import synchronization with the default realm that is installed with Oracle Internet Directory Release 11*g* Release 1 (11.1.1). If you are upgrading from an earlier version of Oracle Internet Directory, or if the synchronization is with a nondefault Oracle Internet Directory realm, then be sure that the necessary privileges in the proper subtree or containers are granted to the import profiles handling the synchronization.

> **See Also:** The chapter about access controls in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*

Customizing an LDIF ACL for an Oracle Internet Directory Back-end Directory

For an ACL template in LDIF format, see the file $*ORACLE_HOME*/ldap/schema/oid/oidRealmAdminACL.sbs. If your Oracle back-end directory is Oracle Internet Directory and you have not changed the ACLs on the default realm, then this template file can be applied directly after instantiating the substitution variables, replacing %s_SubscriberDN% with the default realm DN in Oracle Internet Directory, and replacing %s_OracleContextDN% with cn=OracleContext,*default_realm_DN* respectively.

For example, if realmacl.ldif is the instantiated file, then you can upload it by using the following ldapmodify command:

```
$ORACLE_HOME/bin/ldapmodify -h OID host -p OID port
-D binddn -q -v -f realmacl.ldif
```

After executing the command, you will be prompted for the password for privileged directory user.

17.3.2.2 Customizing ACLs for Export Profiles

To enable the Oracle Directory Integration Platform to access a connected directory, you must create an identity in the connected directory. This identity is configured in each export profile.

To Customize the ACL to Support the Synchronization of User Records Located Outside of the Default Realm

If your Oracle back-end directory is Oracle Internet Directory and you need to synchronize user records located outside of the Oracle Internet Directory default realm, modify the ACL using an LDIF file as follows.

Note: This ACL change is required to export OID user passwords that are outside of the default realm to connected directories using DIP sync.

1. Query the ACIs in the root-directory specific entry, and save the output to an LDIF file as a backup:

```
ldapsearch -h OID host -p port -D cn=orcladmin -w password -s base -L
-b "" objectclass=* orclaci orclentrylevelaci > /tmp/orig-root-acis.ldif
```

2. Find the following ACI:

```
orclaci: access to
attr=(userpkcs12,orclpkcs12hint,userpassword,pwdhistory,orclrevpwd) by
 group="cn=OracleUserSecurityAdmins,cn=Groups,cn=OracleContext"
 (search,read,write,compare) by self (search,read,write,compare) by * (none)
```

3. Modify the ACI to include the DIP DN and save the file as `new-root-acis.ldif`.

   Your modified file should look like this:

```
orclaci: access to
attr=(userpkcs12,orclpkcs12hint,userpassword,pwdhistory,orclrevpwd) by
group="cn=OracleUserSecurityAdmins,cn=Groups,cn=OracleContext"
(search,read,write,compare) by self (search,read,write,compare) by
dn="cn=odisrv,cn=Registered Instances,cn=Directory Integration
Platform,cn=products,cn=oraclecontext" (search,read)* by * (none)
```

4. Add the following two lines after the first line of the file:

```
changetype: modify
replace: orclaci
```

   The top of your file should read as follows:

```
dn: Container DN
changetype: modify
replace: orclaci
orclaci: access to
...
```

5. Use the `ldapmodify` command to apply the modified ACI:

```
$ORACLE_HOME/bin/ldapmodify -h OID host -p OID port -D cn=orcladmin -w password
-v -f /tmp/new-root-acis.ldif
```

17.3.2.3 ACLs for Other Oracle Components

Default ACLs enable you to create, modify, and delete users and groups, but only in the users and groups containers under the default realm. To synchronize objects in other containers, you must customize the ACLs.

There are sample ACL files that you can use to customize ACLs for Oracle Components. These sample files are installed in the *$ORACLE_HOME*`/ldap/schema/oid` directory. They are:

- `oidUserAdminACL.sbs`—Grants necessary rights to the subtree for Oracle components to manage and access users.

- `oidGroupAdminACL.sbs`—Grants necessary rights to the subtree for Oracle components to manage and access groups.
- `oidUserAndGroupAdminACL.sbs`—Grants the privileges for Oracle components to manage and access users and groups in the subtree.

You can customize your ACL policy to grant privileges on a container-by-container basis with the required rights.

> **See Also:** The chapter about access controls in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for instructions on customizing ACLs

17.3.3 Customizing Mapping Rules

Mapping rules, an important part of the synchronization profile, determine the directory information to be synchronized and how it is to be transformed when synchronized. You can change mapping rules at run time to meet your requirements.

Each sample synchronization profile includes default mapping rules. These rules contain a minimal set of default user and group attributes configured for out-of-the-box synchronization.

> **Note:** When a synchronization is underway, it relies on the mapping rules configured prior to any changes in the directory. To ensure consistent mapping, you may need to remove an already synchronized entry or perform a full synchronization.

Mapping rules govern the way data is transformed when a source directory and a destination directory are synchronized. Customize the default mapping rules found in the sample profiles when you need to do the following:

- Change distinguished name mappings. The distinguished name mappings establish how the connected directory DIT maps to the Oracle back-end directory DIT.
- Change the attributes that need to be synchronized.
- Change the transformations (mapping rules) that occur during the synchronization.

You can perform any mapping if the resulting data in the destination directory conforms to the schema in that directory.

> **See Also:**
>
> - The section "Configuring Mapping Rules" on page 6-3 for a full discussion of mapping rules
> - The section "Supported Attribute Mapping Rules and Examples" on page 6-11 for examples of how attribute values are transformed when synchronized from one directory to another
> - The file $*ORACLE_HOME*`/ldap/odi/conf/activeimp.map.master` for an example of import mapping rules

Once you have established a working synchronization between the Oracle back-end directory and a connected directory, you can customize the attribute mapping rules for your synchronization profiles to meet the needs of your deployment.

To customize the attribute mapping rules for your synchronization profiles:

1. Make a duplicate of the sample mapping rules file. The sample mapping rules files are stored in the $*ORACLE_HOME*`/ldap/odi/conf` directory with the extension of `map.master` for the various profiles.
2. Edit the sample mapping rules file to make the previously discussed modifications. You can find instructions for editing mapping rules in "Configuring Mapping Rules" on page 6-3.
3. After the changes are made, use the `update` operation of the `manageSyncProfiles` command to update the profile. For example, the following command updates a profile name `myImportProfile` with a properties file named `myPropertiesFile`:

   ```
   manageSyncProfiles update -profile profile_name -file myPropertiesFile
   ```

   See Also: The manageSyncProfiles section in the Oracle Directory Integration Platform tools chapter of the *Oracle Identity Management User Reference*.

4. Wait until the scheduling interval has elapsed, and then check the synchronized users and groups to ensure that the attribute mapping rules meet your requirements.

   Tip: You may find it helpful to add test users and groups to the Oracle back-end directory or the connected directory when customizing attribute mapping rules.

17.3.4 Configuring the Connected Directory Connector for Synchronization in SSL Mode

By default, SSL is not enabled for the import and export synchronization profiles created with the `expressSyncSetup` command. Whether or not you synchronize in the SSL mode depends on your deployment requirements. For example, synchronizing public data does not require SSL, but synchronizing sensitive information such as passwords does. To synchronize password changes between the Oracle back-end directory and a connected directory, you must use SSL server authentication mode.

Note: Be sure that you can successfully synchronize users in non-SSL mode before attempting to configure your synchronization profiles for SSL.

Securing the channel requires:

- Enabling SSL between the Oracle back-end directory and the Oracle Directory Integration Platform
- Enabling SSL between the Oracle Directory Integration Platform and the connected directory

Although you can enable SSL either between the Oracle back-end directory and the Oracle Directory Integration Platform, or between that server and the connected directory, Oracle recommends that you completely secure the channel before you

synchronize sensitive information. In certain cases, such as password synchronization, synchronization can occur only over SSL.

Configuring SSL requires the following:

- Running the Oracle directory server in SSL mode as described in the chapter on Secure Sockets Layer (SSL) in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.
- Running the Oracle Directory Integration Platform in the SSL mode as described in Chapter 2, "Security Features in Oracle Directory Integration Platform." The SSL mode for Directory Integration Platform must be the same mode used when the the Oracle back-end directory server started. SSL mode 1 is no authentication and SSL mode 2 is server authentication.

  Note: Oracle Directory Integration Platform only supports the No Authentication SSL mode (SSL mode 1) if your Oracle back-end directory is Oracle Internet Directory. If Oracle Unified Directory or Oracle Directory Server Enterprise Edition is your Oracle back-end directory, SSL Server Authentication (SSL mode 2) is your only SSL option.

- Running the connected directory server in SSL mode. Communication with a connected directory over SSL requires SSL server authentication. This requires that both the Oracle back-end directory and the Oracle Directory Integration Platform be run in SSL server authentication mode.

Perform the following steps to configure communication with a connected directory in SSL mode:

1. Generate a certificate for the connected directory. Only the trust point certificate from the server is required. Put the certificate in the connected directory's certificate store.
2. Export the trusted Certificate Authority (CA) certificates to Base 64 encoded format.
3. Import the trusted CA certificates to the Java KeyStore (JKS) using the keytool command. If Oracle Directory Integration Platform is already using an existing JKS, identify the location of it using the `-keystore` *`PATH_TO_JKS`* option. If Oracle Directory Integration Platform does not already have a JKS to use, keytool will create one at the location identified by the `-keystore` *`PATH_TO_JKS`* option.

   For example:

   ```
   keytool -importcert -trustcacerts -alias mycert -file PATH_TO_CERTIFICATE \
   -keystore PATH_TO_JKS
   ```

   If this is the first time you are using the JKS identified by the `-keystore` *`PATH_TO_JKS`* option, you must provide its password and also perform the following steps a and b:

   a. Update the Directory Integration Platform configuration with the location and password used in step 3 by using the `manageDIPServerConfig` command. For example:

      ```
      manageDIPServerConfig set -h HOST -p PORT -D WLS_USER \
      -attribute keystorelocation -value PATH_TO_CERTIFICATE
      ```

b. Update the credential in the Credential Store Framework (CSF) using the following WLST command and replacing the *PASSWORD* variable with the password used when the keystore was created:

```
createCred(map="dip", key="jksKey", user="jksUser",
password="PASSWORD",desc="jks password")
```

4. Modify the connected directory connection information, including the host name, profile, and `connectedDirectoryURL` attribute, using the modify operation of the manageSyncProfiles command.

```
manageSyncProfiles update -profile profile_name -file myMapFile
```

When you configure the `connectedDirectoryURL` attribute, use the following format:

```
host:port:sslmode
```

Supported values for `sslmode` are as follows:

Table 17–1 Supported Values for sslmode in connectedDirectoryURL Attribute

Supported sslmode Value	Description
0	No SSL mode. Supported for all directory types.
1	No Authentication mode. No certificate. Supported only for Oracle Internet Directory.
2	Server-Only Authentication mode. Requires certificate. Supported for all directory types.

5. If you used a new JKS in step 3, you must restart the Oracle Directory Integration Platform in SSL mode. If you used an existing JKS in step 3, go to step 6 now.

6. Add a test user and verify that it synchronizes successfully. If the test user does not synchronize successfully, then troubleshoot your SSL configuration.

Note: The Oracle Directory Integration Platform does not support SSL in client/server authentication mode.

See Also: "Managing the SSL Certificates of Back-End Directories and Connected Directories" on page 4-14

17.3.5 Enabling Password Synchronization from the Oracle Back-end Directory to a Connected Directory

Password synchronization to a connected directory from an Oracle Unified Directory back-end directory or an Oracle Directory Server Enterprise Edition back-end directory is not supported.

To synchronize passwords from Oracle Internet Directory to a connected directory, you must enable the password policy and you may have to enable reversible password encryption in the Oracle Internet Directory server.

Enable reversible password encryption in the Oracle Internet Directory server *only* if the hashing algorithm between Oracle Internet Directory and the connected directory is incompatible or unsupported.

For example, IBM Tivoli Directory Server and Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server) support similar hashing algorithms as Oracle Internet Directory. Therefore, to synchronize passwords from Oracle Internet Directory to IBM Tivoli Directory Server or Oracle Directory Server Enterprise Edition, you must enable *only* the password policy in the Oracle Internet Directory server.

However, to synchronize passwords from Oracle Internet Directory to Microsoft Active Directory or Novell eDirectory, which both do not support similar hashing algorithms as Oracle Internet Directory, you must enable the password policy *and* reversible password encryption in the Oracle Internet Directory server.

Note: As of Oracle Internet Directory 10*g* (10.1.4.0.1), Oracle Internet Directory supports multiple password policies in each realm, commonly known as Fine-Grained Password Policies.

Refer to the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for more information about Fine-Grained Password Policies.

To enable the password policy, assign a value of `1` to the `orclPwdPolicyEnable` attribute in the appropriate container. To enable reversible password encryption in the Oracle Internet Directory server, assign a value of `1` to the `orclpwdEncryptionEnable` attribute in the appropriate container.

For example, to enable the password policy and reversible password encryption on the default policy for a realm, assign a value of `1` to the `orclPwdPolicyEnable` and `orclpwdEncryptionEnable` attributes in the following entry:

```
cn=default,cn=PwdPolicyEntry,cn=common,cn=products,cn=oraclecontext,Realm_DN
```

You can do this by using `ldapmodify` and uploading an LDIF file containing the following entries:

```
dn: cn=default,cn=PwdPolicyEntry,cn=common,cn=products,cn=oraclecontext,Realm_DN
changetype: modify
replace: orclpwdpolicyenable
orclpwdpolicyenable: 1
-
replace: orclpwdencryptionenable
orclpwdencryptionenable: 1
```

See Also: *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information on managing Oracle Internet Directory password policies.

17.3.6 Configuring External Authentication Plug-ins

Oracle Directory Integration Platform supports Java-based external authentication plug-ins. Oracle recommends that you use the Java plug-ins instead of the older, PL/SQL-based plug-ins, which only support Microsoft Active Directory and Oracle Directory Server Enterprise Edition / Sun Java System Directory Server.

The configuration tool for the plug-ins is a Java program called `oidexcfg`. You use it to configure Java-based external authentication plug-ins for Microsoft Active Directory, Oracle Directory Server Enterprise Edition (Sun Java System Directory Server), Novell eDirectory, IBM Tivoli Directory Server, and OpenLDAP.

> **Note:** The `oidexcfg` tool configures an external authentication plug-in to work only with a single domain. You must perform the steps described in "Configuring External Authentication Against Multiple Domains" to set up an external authentication plug-in to work with multiple domains.

To configure an external authentication plug-in, perform the following steps:

1. (*Optional*) Perform this step only if you want to use SSL to secure the communication between the authentication plug-in and the external LDAP directory. If you do not want to secure the communication, proceed to step 2 now.

   To secure the communication between the authentication plug-in and the external LDAP directory using SSL, a trusted certificate from the external, authenticating directory must reside in a wallet on the file system. When you configure the plug-in using `oidexcfg` in step 3, you will be prompted to enter information about the external LDAP directory configuration and you can identify the location of this wallet.

   If you want to use SSL, put the certificate in a new or existing wallet now.

   > **Note:** The certificate enables SSL to secure the communication between the authentication plug-in and the external LDAP directory—it does not secure the communication with the Oracle back-end directory when you execute `oidexcfg` in step 3.

2. Include `oidexcfg.jar` and `ldapjclnt11.jar` in the Java CLASSPATH environment variable. To set the environment variable:

   In UNIX/Linux environments:

   ```
   setenv CLASSPATH=$ORACLE_HOME/ldap/jlib/oidexcfg.jar:$ORACLE_
   HOME/ldap/jlib/ldapjclnt11.jar:$CLASSPATH
   ```

   In Windows environments:

   ```
   set CLASSPATH=%ORACLE_HOME%/ldap/jlib/oidexcfg.jar;%ORACLE_
   HOME%/ldap/jlib/ldapjclnt11.jar;%CLASSPATH%
   ```

3. Configure the plug-in using `oidexcfg` by executing the following command. You will be prompted to enter information about the external LDAP directory configuration, including the location of the wallet containing the trusted certificate required for SSL.

   > **Note:** You must identify the location of the wallet file using a fully-qualified path, for example:
   >
   > ```
   > /etc/ORACLE_HOME/wallets/ewallet.p12
   > ```

   Execute the following command to configure the plug-in using `oidexcfg`:

   ```
   java -classpath $CLASSPATH oracle.ldap.extplg.oidexcfg -h OID_Host
   -p OID_Port -D BindDN -w password -t Directory_Type
   ```

   The -t option that identifies the directory type supports the following values:

- `ad` for Microsoft Active Directory
- `adam` for Microsoft Active Directory Application Mode
- `iplanet` for Oracle Directory Server Enterprise Edition and Sun Java System Directory Server
- `edirectory` for Novell eDirectory
- `openldap` for OpenLDAP
- `tivoli` for IBM Tivoli Directory Server

17.3.6.1 Configuring External Authentication Against Multiple Domains

To set up an external authentication plug-in to work with multiple external authentication domains, you must perform some manual instructions after you run the external configuration tool. Proceed as follows:

1. Configure the external authentication plug-in as described in "Configuring External Authentication Plug-ins".
2. Search for the plug-in configuration entries created by the configuration tool in step 1, and redirect the search output to a file. Use an `ldapsearch` command similar to this:

```
ldapsearch -p 3060 -D binddn -q -s sub -L \
    -b "cn=plugin,cn=subconfigsubentry" cn="oidexplg_*_ad" >> output.ldif
```

Note: You will be prompted for the password.

The example shows an Microsoft Active Directory `cn`. Use the correct plug-in `cn` for the type of plug-in you configured, as shown in Table 17–2. You can use `*` as a wildcard, as shown in the example.

Table 17–2 Distinguished Names of External Authentication Plug-ins

Plug-in Type	DN
Microsoft Active Directory	`cn=oidexplg_compare_ad, cn=plugin,cn=subconfigsubentry` `cn=oidexplg_bind_ad, cn=plugin,cn=subconfigsubentry`
Oracle Directory Server Enterprise Edition (Sun Java System Directory Server)	`cn=oidexplg_compare_iplanet, cn=plugin,cn=subconfigsubentry` `cn=oidexplg_bind_iplanet, cn=plugin,cn=subconfigsubentry`
Novell eDirectory	`cn=oidexplg_compare_Novell eDirectory, cn=plugin,cn=subconfigsubentry` `cn=oidexplg_bind_Novell eDirectory, cn=plugin,cn=subconfigsubentry`
OpenLDAP	`cn=oidexplg_compare_openldap, cn=plugin,cn=subconfigsubentry` `cn=oidexplg_bind_openldap, cn=plugin,cn=subconfigsubentry`

3. Examine the output file. For an Microsoft Active Directory plug-in, the output file resembles the following:

```
dn: cn=oidexplg_compare_ad,cn=plugin,cn=subconfigsubentry
cn: oidexplg_compare_ad
objectclass: orclPluginConfig
objectclass: top
orclpluginname: oidexplg.jar
orclplugintype: operational
orclpluginkind: Java
orclplugintiming: when
orclpluginldapoperation: ldapcompare
orclpluginsecuredflexfield;walletpwd: password
orclpluginsecuredflexfield;walletpwd2: password
orclpluginversion: 1.0.1
orclpluginisreplace: 1
orclpluginattributelist: userpassword
orclpluginentryproperties:
(!(&(objectclass=orcladobject)(objectclass=orcluserv2)))
orclpluginflexfield;host2: host.domain.com
orclpluginflexfield;port2: 636
orclpluginflexfield;isssl2: 1
orclpluginflexfield;host: host.domain.com
orclpluginflexfield;walletloc2: /location/wallet
orclpluginflexfield;port: 389
orclpluginflexfield;walletloc: /tmp
orclpluginflexfield;isssl: 0
orclpluginflexfield;isfailover: 0
orclpluginclassreloadenabled: 0
orclpluginenable: 0
orclpluginsubscriberdnlist: cn=users,dc=us,dc=oracle,dc=com

dn: cn=oidexplg_bind_ad,cn=plugin,cn=subconfigsubentry
cn: oidexplg_bind_ad
objectclass: orclPluginConfig
objectclass: top
orclpluginname: oidexplg.jar
orclplugintype: operational
orclpluginkind: Java
orclplugintiming: when
orclpluginldapoperation: ldapbind
orclpluginversion: 1.0.1
orclpluginisreplace: 1
orclpluginentryproperties:
(!(&(objectclass=orcladobject)(objectclass=orcluserv2)))
orclpluginclassreloadenabled: 0
orclpluginflexfield;walletloc2: /location/wallet
orclpluginflexfield;port: 389
orclpluginflexfield;walletloc: /tmp
orclpluginflexfield;isssl: 0
orclpluginflexfield;isfailover: 0
orclpluginflexfield;host2: host.domain.com
orclpluginflexfield;port2: 636
orclpluginflexfield;isssl2: 1
orclpluginflexfield;host: host.domain.com
orclpluginenable: 0
orclpluginsecuredflexfield;walletpwd: password
orclpluginsecuredflexfield;walletpwd2: password
orclpluginsubscriberdnlist:
cn=users,dc=us,dc=oracle,dc=com
```

4. Create a new LDIF file from the output file as follows:
   a. Change the entry names. In the example shown in the previous step, you would change `cn=oidexplg_compare_ad,cn=plugin,cn=subconfigsubentry` to `cn=oidexplg_compare_`**`ad1`**`,cn=plugin,cn=subconfigsubentry` and `cn=oidexplg_bind_ad,cn=plugin,cn=subconfigsubentry` to `cn=oidexplg_bind_`**`ad1`**`,cn=plugin,cn=subconfigsubentry`.
   b. Change the value for `orclpluginenable`. Use value `1` if you want to enable it, and use value `0` if you want to disable it.
   c. Change the values for `orclpluginflexfield;host` and `orclpluginflexfield;port` for the external directory host name and port number.
   d. Change the value for `orclpluginflexfield;isssl`. Use value `1` if you want to enable the SSL connection against the external directory, and use value `0` if you want to disable. If you use value `1`, you will also need to change the value of `orclpluginflexfield;walletloc` and `orclpluginsecuredflexfield;walletpwd` for the wallet location and password.
   e. Change `orclpluginflexfield;isfailover`. Use value `1` if to set up the failover against a backup external directory. If you use value `1`, then you must also change the value of `orclpluginflexfield;host2`, `orclpluginflexfield;port2` for the host name and port number. To use an SSL connection against the backup directory server, you must to change the value for `orclpluginflexfield;walletloc2` and `orclpluginsecuredflexfield;walletpwd2`.
   f. Modify `orclpluginsubscriberdnlist` for the plug-in invocation naming context.
   g. Modify `orclPluginRequestGroup` for the plug-in request group. If this attribute is missing in the search out put, then just add the attribute and value in the LDIF file.
5. Add the modified plug-in configuration entries to the Oracle Internet Directory server. Use a command similar to the following:

```
$ORACLE_HOME/ldap/bin/ldapadd -h host -p port -D binddn -q \
      -v -f input.ldif
```

Note: You will be prompted for the password.

17.4 Writing Custom Synchronization Connectors

Oracle Directory Integration Platform supports custom synchronization connectors. This topic provides information to help you write custom connectors and contains the following sections:

- Inbound Connectors
- Outbound Connectors

17.4.1 Inbound Connectors

Perform the following steps to write an inbound connector:

1. Implement the Reader. The Reader generally extends the target system connector class and implements the DISReadInterface. The different methods of the DISReadInterface are specified in its the javadoc. Refer to "Sample Reader" to see an example Reader implementation.
2. Create a sample config file. The following is a typical config file:

```
[INTERFACEDETAILS]
Reader: Complete_classname_including_packageName
SkipErrorToSyncNextChange: false
SearchDeltaSize: 500
UpdateSearchCount: 100
```

3. Create a mapfile containing a set of mapping rules.
4. Create a properties file by setting the configfile, mapfile, and filter parameters.

To test the inbound connector:

1. Create a test profile using the register operation of the manageSyncProfiles command. Refer to "Managing Synchronization Profiles Using manageSyncProfiles" on page 7-16 for more information.
2. Verify your logging messages.
3. Verify synchronization occurred by examining Oracle Internet Directory to see if the appropriate entries were created.

17.4.1.1 Sample Reader

```
package oracle.ldap.odip.gsi;
import oracle.ldap.odip.engine.AttrHandler;
import oracle.ldap.odip.engine.ChangeRecord;
import oracle.ldap.odip.engine.Connector;
import oracle.ldap.odip.engine.ConfigReader;
import oracle.ldap.odip.engine.Constants;
import oracle.ldap.odip.engine.DISReadInterface;
import oracle.ldap.odip.engine.DISFilterInterface;
import oracle.ldap.odip.engine.ODIException;
import oracle.ldap.odip.engine.Debug;
import oracle.ldap.odip.map.MapRules;
import oracle.ldap.odip.map.OrclFilter;
import oracle.ldap.odip.util.Utils;
 //Imports added for ODLLogger
import oracle.core.ojdl.logging.ODLLogger;
import oracle.dms.context.ExecutionContext;
import oracle.core.ojdl.logging.ODLLevel;
import oracle.core.ojdl.logging.ODLHandler;
import java.util.logging.Handler;
import java.util.logging.Level;

import oracle.ldap.odip.DIPLogger;

public class SampleReader implements DISReadInterface
{

  /*
  ** Member variables used
  */
  protected  NamingEnumeration  mEnumerate;
  protected  Attributes         mAttribs;
  protected  Attribute          mAttrib;
```

```
protected  Attribute          mAttribAllValues;
protected  SearchResult       mResult;
protected  MapRules           mMapRules;
/*
** Vector to store the list of required attributes
*/
protected  Vector             mReqAttrList = new Vector();

/*
** List of source attributes whose changes need to be mapped
*/
protected  Vector             mSrcAttrList = new Vector();
protected  String             mMapFilter;
protected  int                mAppliedChangeNum = 0;
protected  int                mAvailableChangeNum = 700;
protected  DISFilterInterface mFilter;

/*
** LastChangeNumber that is read
*/

protected  String            mReadChangeNum;

/*
** List of attributes to be returned in changelog LDAPSearch
*/
protected  String[]           mRetAttribs;
private    int                mErrorCode = 0;


/*
** Constructor
*/
public SampleReader()
{
}

/**
** Constructor with the connector
*/
public SampleReader( Connector conn )
{
  super(conn);

}

/**
** Get the last change key value
**
* @param   boolean Operation is success/failure
* @return  Object  lastkeyvalue to be stored
*/
public Object getLastChangeKey(boolean val)
{
    if ( val == false )
    {
        int nval = Integer.parseInt(mReadChangeNum);
        if ( nval > 0 )
        {
          nval--;
```

```
            }
            mReadChangeNum = String.valueOf(nval);
        }
        return (mReadChangeNum);
    }

    /**
    ** Initializes required values from hashtable passed from Profile
    **
    ** @param   Connector   connection details with credentials
    ** @param   Hashtable   with the required parameters
    ** @throws  ODIException Indicating connection failure
    */
    public void initialise(Connector conn,Hashtable pHash)
        throws ODIException
    {
         m_logger.finest ( "Entry: SampleReaders.initialise");
        setValues(conn);
        mMapRules = (MapRules)pHash.get(Constants.MAPRULE_STR);

        readCtx = connect();

        pHash.put("READCONTEXT", readCtx);
        pHash.put(Constants.READERCHANGEKEY_STR, Constants.CHANGE_NUM);
        String key = (String)pHash.get(Constants.LASTAPPLIEDCHG_STR);
        String val = null;
        if ( key != null )
            val = (String)pHash.get(key);
        if ( val != null )
            mAppliedChangeNum = Integer.parseInt((String)pHash.get(key));
        mReadChangeNum = (String)pHash.get(key);
        pHash.put(key, mReadChangeNum);
        mFilter = (DISFilterInterface)pHash.get(Constants.MATCHRULE_STR);
        mAvailableChangeNum = Integer.parseInt(initAvailableChgKey());
        mSaveLastChgNum = mAppliedChangeNum;

        try {
          SearchControls pControls = new SearchControls();
          pControls.setSearchScope(SearchControls.OBJECT_SCOPE);
          pControls.setReturningAttributes(mRetAttribs);
          pControls.setTimeLimit(3000000);

mEnumerate = mLdapCtx.search("","objectclass=*",pControls);
          while ( mEnumerate.hasMoreElements() )
          {
            mResult = (SearchResult)mEnumerate.nextElement();
            mAttribs = mResult.getAttributes();
          }
          // END INFEASIBLE
          ConfigReader configInfo = (ConfigReader) pHash.get(Constants.CONFINFO_
STR);

          if (configInfo != null) {
              mUpdateSearchCount = configInfo.getUpdateSearchCount();
              mSearchDelta = configInfo.getSearchDeltaSize();
          }
        } catch (Exception ex)
        // BEGIN INFEASIBLE
        {
```

```
            throw new ODIException(ODIException.LDAP_INITIALIZATION_EXCEPTION,ex);
          }
          // END INFEASIBLE
          m_logger.finest ( "Exit: SampleReaders.initialise");
      }

      /**
      ** Search the changelog
      ** @throws ODIException
      */
      public int searchChanges()
          throws ODIException
      {

            int  temp;
            int searchDelta = (int) mSearchDelta;
            if ( mAvailableChangeNum <= mAppliedChangeNum ) return -1;
            int  minChgNum = mAppliedChangeNum+1;
            if ( mAvailableChangeNum -  mAppliedChangeNum >= searchDelta)
                temp = mAppliedChangeNum + searchDelta;
            else
                temp = mAvailableChangeNum;

            String searchF = "";
            if ( mFilter != null ) {
              searchF = mFilter.getSearchFilter();
              m_logger.log(ODLLevel.NOTIFICATION,"SEARCHF", searchF );
            }

            StringBuffer filter = new StringBuffer(300);

            /**
             * SearchChanges is called to get all changes
             *
             */
            try {

               mEnumerate = mReadCtx.search(
                                          filter.toString());

            }
            catch ( Exception ex )
            // BEGIN INFEASIBLE
            {
                throw ( new ODIException(ODIException.LDAP_SEARCH_EXCEPTION,
                                        ex) );
            }
            finally {

            m_logger.log(ODLLevel.NOTIFICATION, "SEARCH_SUCCESSFUL" ,new Integer( temp
));

            mAppliedChangeNum = temp;
          return mErrorCode;
      }

      public boolean hasMore()
          throws ODIException
      {
        boolean  retval = false;
```

```
  int count =0;

  try {
    if ( mEnumerate.hasMoreElements() )
    {
      retval = true;
    }
    else
    {
      while ( mAvailableChangeNum > mAppliedChangeNum ) {
        if ( count >= mUpdateSearchCount )
            break;

        searchChanges();
count++;
        if (mEnumerate.hasMoreElements())
        {
          retval = true;
          break;
        }
else
  mReadChangeNum = String.valueOf(mAppliedChangeNum);
      }
    }
  }
  catch( Exception ex )
  // BEGIN INFEASIBLE
  {
    throw (new ODIException(ODIException.LDAP_HASMORE_EXCEPTION,ex));
  }
  // END INFEASIBLE
  if (retval == false) {  // no more results
    mReadChangeNum = (new Integer(mAvailableChangeNum)).toString();
  }
  return retval;
}


 /**
** Read the next change from the source
**
** @return  Object  the header part of the changes read.
*/
public Object getNextChange()
    throws ODIException
{
  try {

    if ( mEnumerate.hasMoreElements() )
    {
      mResult = (SearchResult)mEnumerate.nextElement();
      mAttribs = mResult.getAttributes();

  }
  catch ( Exception e )
  // BEGIN INFEASIBLE
  {
    throw (new ODIException (ODIException.LDAP_GETNEXT_EXCEPTION, e));
  }
  // END INFEASIBLE
```

```
    return mAttribs;
  }

  /**
  ** Create the change record from the data read from the file.
  **
  ** @returns  ChangeRecord
  */
  public ChangeRecord createChangeRecord(String dn)
      throws ODIException
  {

      // Create the changerecord based on the mAttribs which contains all the
attributes.

  }

  public String initAvailableChgKey() throws ODIException
  {
      // set the available changekey value. This reads the value equivalent to the
latest changelog number in the ldap world.

   }

}
```

17.4.2 Outbound Connectors

Perform the following steps to write an outbound connector:

1. Implement the Writer. The Writer generally extends the target system connector class and implements the DISWriteInterface. The different methods of the DISWriteInterface are specified in its Javadoc. Refer to "Sample Writer" to see an example Reader implementation.
2. Create a sample config file. The following is a typical config file:

```
[INTERFACEDETAILS]
Reader: Complete_classname_including_packageName
SkipErrorToSyncNextChange: false
SearchDeltaSize: 500
UpdateSearchCount: 100
```

3. Create a mapfile containing a set of mapping rules.
4. Create a properties file by setting the configfile, mapfile, and filter parameters.

To test the outbound connector:

1. Create a test profile using the register operation of the manageSyncProfiles command. Refer to "Managing Synchronization Profiles Using manageSyncProfiles" on page 7-16 for more information.
2. Verify your logging messages.
3. Verify synchronization occurred by examining Oracle Internet Directory to see if the appropriate entries were created.

17.4.2.1 Sample Writer

```
 */



import oracle.ldap.odip.engine.AttrHandler;
import oracle.ldap.odip.engine.ChangeRecord;
import oracle.ldap.odip.engine.ConfigReader;
import oracle.ldap.odip.engine.Connector;
import oracle.ldap.odip.engine.Constants;
import oracle.ldap.odip.engine.DISWriteInterface;
import oracle.ldap.odip.engine.ODIException;
import oracle.ldap.odip.map.MapRules;
import oracle.ldap.odip.util.Utils;

import oracle.core.ojdl.logging.ODLLogger;
import oracle.core.ojdl.logging.ODLLevel;
import oracle.core.ojdl.logging.ODLHandler;
import java.util.logging.Handler;
import java.util.logging.Level;

import oracle.ldap.odip.DIPLogger;


public class SampleWriter  implements DISWriteInterface {
    protected Hashtable mProfile;
    protected int mErrorCode = 0;
    protected String mLastKeyValue;
    protected String mLastWrittenKey;
    protected Vector mWriteFilter = new Vector();
    protected MapRules mMapRules;
    protected String mNamingContext = "";
    private String mOrigDstDn = "";
    protected boolean mHandleModAsAdd = false;

    /* Constructor */
    public LDAPWriter() {
    }

    public LDAPWriter(Connector conn) {
        super(conn);
    }

    public void initialise(Connector conn, Hashtable pHash)
        throws ODIException {
    m_logger.finest("Entry: LDAPWriter.initialise");
        setValues(conn);
        mProfile = pHash;
        mMapRules = (MapRules) pHash.get(Constants.MAPRULE_STR);

        connect();
       ConfigReader configInfo = (ConfigReader) pHash.get(Constants.CONFINFO_STR);

        if (configInfo != null) {
            //mSearchDelta = configInfo.getSearchDeltaSize();
            mHandleModAsAdd = configInfo.getHandleModAsAdd();
        }

        mLastWrittenKey = (String) pHash.get(Constants.READERCHANGEKEY_STR);
        pHash.put("WRITECONTEXT", mLdapCtx);
```

```
        NamingEnumeration filter = (NamingEnumeration) pHash.get("WriteFilter");

        try {
            while (filter.hasMoreElements()) {
                mWriteFilter.add((String) filter.next());
            }
        } catch (Exception ex) {
            //System.out.println("Error in initializing filter");
        }

        /*
        ** Get the lastapplied changekey value from the profile
        ** and use that string to determine the 'lastappliedchangenum'
        ** or lastappliedchangetime to be stored as the 'lastkeyvalue'
        **
        ** Each of the insert/modify/delete routines, if the operation is
        ** successful, that lastkeyvalue is updated correspondingly. Otherwise
        ** it has the previous successful operation value
        */
m_logger.finest ( "Exit: LDAPWriter.initialise" );
    }

    public void setChanges(ChangeRecord chgrec) {
        mChanges = chgrec;
    }

    public ChangeRecord getChanges() {
        return mChanges;
    }

    public String getLastChangeKey() {
        return mLastKeyValue;
    }

    public int writeChanges() throws ODIException {
    m_logger.finest("Entry: LDAPWriter.writeChanges");
        mErrorCode = 0;

        m_logger.log(ODLLevel.FINE,
            "\n Output ChangeRecord " + mChanges.toString());

        String dn = mChanges.getChangeKey();

       if ( mHandleModAsAdd && (mChanges.getChangeType() == Constants.CHGTYPE_
MODIFY)) {
            try {
              mLdapCtx.getAttributes( mChanges.getChangeKey() );

            }
            catch (NameNotFoundException nnfe) {
              m_logger.log(ODLLevel.ERROR,"ERROR_DN_CONN_DIR");
              mChanges.setChangeType(Constants.CHGTYPE_MODRADD);
            }
            catch (NamingException ne) {
              m_logger.log(ODLLevel.ERROR,"LDAP_WNAMING_EXCEPTION" , ne);
            }

       }
        m_logger.log(ODLLevel.FINE,
```

```
                    "Changetype is " + mChanges.getChangeType());
            mChanges.setChangeKey(ndn);

            if (dn.length() > 1) {
                //testnew(dn);
                switch (mChanges.getChangeType()) {
                case Constants.CHGTYPE_ADD:

                    if (mChanges.size() > 0) {
                        insert();
                    }

                    break;

                case Constants.CHGTYPE_MODIFY:
                    // non-changelog-based changes
                    if (mChanges.size() > 0) {
                                          modify();
                    }
else {
    mErrorCode = -1;
}

                    break;

                case Constants.CHGTYPE_DELETE:
                    delete();

                    break;

                case Constants.CHGTYPE_MODRADD:

                    // non-changelog-based changes
                    if (mChanges.size() > 0) {
                        modifyRadd();
                    }

                    break;

                case Constants.CHGTYPE_MODRDN:
                    modRDNchangelog(dn);

                    break;

                case Constants.CHGTYPE_MODDN:
                    m_logger.log(ODLLevel.FINE,
                                 "Processing moddn");
                    modDNchangelog(dn);
                    break;

                default:

                    //INFEASIBLE
                    break;
                }
            } else // BEGIN INFEASIBLE
             {
                m_logger.log(ODLLevel.ERROR,
                    "ENTRY_NOT_EXISTS_DELETE");
                m_logger.log(ODLLevel.FINE,
```

```
                "Synchrozing a deletion, entry to delete is not found. Ignore.");
            mErrorCode = 99;

            return mErrorCode;
        }

        // END INFEASIBLE
        Object chgInfo = mChanges.getChangeInfo();

        try {
            if (chgInfo instanceof Attributes) {
                Attributes attrs = (Attributes) chgInfo;
                mLastKeyValue = (String) ((Attribute)
attrs.get(mLastWrittenKey)).get();
            }
        } catch (Exception ex) {
            //System.out.println("Caught the exception here " + mErrorCode);
            if (mErrorCode != 0) {
                m_logger.log(ODLLevel.ERROR,
                    "EXCEPTION_FOR_DN", new Object [] { dn, new Integer (
mErrorCode ) , ex.toString()});
            }
        }

        mChanges.setChangeKey(mOrigDstDn);

        return mErrorCode;
    }

    public void insert() throws ODIException {
    m_logger.finest("Entry: LDAPWriter.insert");
        String dn = mChanges.getChangeKey();
        Enumeration attrdtls = mChanges.getAll();

        m_logger.log(ODLLevel.FINE,
            "Processing Insert Operation ..");

        while (attrdtls.hasMoreElements()) {
            AttrHandler temp = (AttrHandler) attrdtls.nextElement();
            attr = attrHandlerToAttr((AttrHandler) temp);
            if (attr != null && temp.getAttrChgType() != Constants.ATTRCHGTYPE_
DELETE) {
                attrs.put(attr);
            }
        }
        createEntry(dn, attrs);
        m_logger.finest("Exit: LDAPWriter.insert");
    }

    public void modify() throws ODIException {
    m_logger.finest("Entry: LDAPWriter.modify");
        String attrname = mChanges.getChangeKey();

        m_logger.log(ODLLevel.FINE,
            "Processing Modify Operation ..");

        int pos = attrname.indexOf('=');

        String naming = null;
```

```
        if (pos > 0) {
            naming = attrname.substring(0, pos).trim();
        }

    }


    /**
    * Delete the entry
    */
    public void delete() {
    m_logger.finest("Entry: LDAPWriter.delete");
        try {
            m_logger.log(ODLLevel.FINE,
                "Processing Delete Operation ..");

}

    /**
    ** Handle the ModRDN operation
    **
    ** @throws ODIException
    */
    protected void modDNchangelog(String newDn) throws ODIException {
      String newDN = null;
      m_logger.log(ODLLevel.FINE,
            "Processing change log based ModRDN operation .." +
            " DN passed in: " + newDn);

      String dn = mChanges.getChangeKey();


    }

    /**
    ** Handle the ModRDN operation
    **
    ** @throws ODIException
    */
    protected void modRDNchangelog(String newDn) throws ODIException {

    }

    protected void performModDN(String oldDN, String newDN)
        throws ODIException {
    }

    /*
    ** First check whether the 'dn' already exists.
    ** If exists,
    **     do a modify.
    ** else
    **     construct objectclasses and do a add
    */

    // public void modifyRadd(boolean rdn) throws ODIException
    public void modifyRadd() throws ODIException {
    m_logger.finest("Entry: LDAPWriter.modifyRadd");

    }
```

```
    /**
    ** Compare the value with the old value, and replace it, if the new value
    ** is different from the old value
    */
    public void checkNReplace(String dn, Attributes attrs)
        throws ODIException {

    }

    //BEGIN INFEASIBLE
    public int getErrorCode() {
        return mErrorCode;
    }


    public int getChangeType() {
        return mChanges.getChangeType();
    }

    public String getEventType() {
        return "";
    }

    //END INFEASIBLE
}
```

18

Integrating with Microsoft Active Directory

This chapter outlines the procedures for integrating Oracle Identity Management with Microsoft Active Directory in a production environment. It contains these topics:

Note: Before continuing with this chapter, you should be familiar with the concepts presented in previous chapters. The following chapters in particular are important:

- Chapter 1, "Introduction to Oracle Identity Management Integration"
- Chapter 4, "Managing the Oracle Directory Integration Platform"
- Chapter 5, "Understanding the Oracle Directory Synchronization Service"
- Chapter 16, "Connected Directory Integration Concepts and Considerations"

If you are configuring a demonstration of integration with Microsoft Active Directory, then see the Oracle By Example series for Oracle Identity Management Release 11*g* Release 1 (11.1.1), available on Oracle Technology Network at
`http://www.oracle.com/technology/`

18.1 Verifying Synchronization Requirements for Microsoft Active Directory

Before configuring basic or advanced synchronization with Microsoft Active Directory, ensure that your environment meets the necessary synchronization requirements by following the instructions in "Verifying Synchronization Requirements" on page 17-1.

18.2 Configuring Basic Synchronization with Microsoft Active Directory

You can use Oracle Enterprise Manager Fusion Middleware Control or the `manageSyncProfiles` command to configure synchronization profiles for Microsoft Active Directory. Refer to Chapter 7, "Managing Directory Synchronization Profiles" for more information.

> **Tip:** Oracle Directory Integration Platform can synchronize one Microsoft Active Directory (AD) with multiple Oracle directory servers at the same time.

18.3 Configuring Advanced Integration with Microsoft Active Directory

When you install Oracle Directory Integration Platform, sample import and export synchronization profiles are automatically created for each of the supported connected directories. The sample synchronization profiles created for Microsoft Active Directory are:

- `ActiveImport`—The profile for importing changes from Microsoft Active Directory to the Oracle back-end directory by using the DirSync approach
- `ActiveChgImp`—The profile for importing changes from Microsoft Active Directory to the Oracle back-end directory by using the USN-Changed approach
- `ActiveExport`—The profile for exporting changes from the Oracle back-end directory to Microsoft Active Directory

---

Notes:

- Whether you use `ActiveImport` or `ActiveChgImp` depends on the method you chose for tracking changes, either DirSync or USN-Changed.
- If you establish integration between Active Directory and the Oracle back-end directory for both exporting and importing users, then you must customize the `ActiveExport` search filter to prevent Oracle Directory Integration Platform from exporting or importing users twice. The following is an example of a customized `ActiveExport` search filter that may be used when both export and import operations are enabled for the same Active Directory instance:

```
odip.profile.condirfilter ="searchfilter=(|(objectclass=
group)(objectclass= organizationalunit)(&(objectclass=user)
(!(objectclass=computer))))"
```

See Also: "Step 3: Customizing the Search Filter to Retrieve Information from Microsoft Active Directory" on page 18-3 for information on customizing the search filter

You can also use the `expressSyncSetup` command or Oracle Enterprise Manager Fusion Middleware Control to create additional synchronization profiles. The import and export synchronization profiles created during the install process or with `expressSyncSetup` are only intended as a starting point for you to use when deploying your integration of the Oracle back-end directory and Microsoft Active Directory. Because the default synchronization profiles are created using predefined assumptions, you must further customize them for your environment by performing the following steps in the order listed:

- Step 1: Planning Your Integration
- Step 2: Configuring the Realm
- Step 3: Customizing the Search Filter to Retrieve Information from Microsoft Active Directory
- Step 4: Customizing the ACLs
- Step 5: Customizing Attribute Mappings
- Step 6: Synchronizing with Multiple Microsoft Active Directory Domains
- Step 7: Synchronizing Deletions from Microsoft Active Directory
- Step 8: Synchronizing in SSL Mode
- Step 9: Synchronizing Passwords
- Step 10: Configuring the Microsoft Active Directory External Authentication Plug-in
- Step 11: Performing Post-Configuration and Administrative Tasks

18.3.1 Step 1: Planning Your Integration

Plan your integration by reading Chapter 16, "Connected Directory Integration Concepts and Considerations", particularly "Microsoft Active Directory Integration Concepts" on page 16-20. Be sure to create a new profile by copying the existing Active Directory template profile by following the instructions in "Creating Synchronization Profiles" on page 7-1.

18.3.2 Step 2: Configuring the Realm

If your Oracle back-end directory is Oracle Internet Directory, configure the realm by following the instructions in "Configuring the Realm" on page 17-8.

18.3.3 Step 3: Customizing the Search Filter to Retrieve Information from Microsoft Active Directory

By default, Microsoft Active Directory Connector retrieves changes to all objects in the container configured for synchronization. If you are interested in retrieving only a certain type of change, for example only changes to users and groups, then you should configure an LDAP search filter. This filter screens out changes that are not required when Microsoft Active Directory Connector queries Microsoft Active Directory. The filter is stored in the `searchfilter` attribute in the synchronization profile.

In the sample profiles `activeChgImp` and `activeImport`, only groups and users are retrieved from Microsoft Active Directory. Computers are not retrieved. The value of the `searchfilter` attribute is set as:

```
searchfilter=(|(objectclass=group)(&(objectclass=user)(!(objectc
lass=computer))).
```

You can use Oracle Enterprise Manager Fusion Middleware Control to customize the search filter.

To customize the search filter by using the Oracle Enterprise Manager Fusion Middleware Control:

1. Open a Web browser and enter the Oracle Enterprise Manager Fusion Middleware Control URL for your environment. The format of the Oracle Enterprise Manager Fusion Middleware Control URL is: https://*host*:*port*/em.
2. Log in to Oracle Enterprise Manager Fusion Middleware Control.
3. In the navigation panel on the left, click or expand the **Identity and Access** entry and then select the **DIP** component that contains the search filter you want to customize.
4. Click the **DIP Server** menu, point to **Administration**, and then click **Synchronization Profiles**. The Manage Synchronization Profiles Page appears.
5. On the Manage Synchronization Server page, select an existing profile and click **Edit**. The Edit Synchronization Profile page appears, opened to the General tab.
6. On the Edit Synchronization Profile page, select the **Filtering** tab.
7. In the Mapping tab page, in the Destination Matching Filter (`orclODIPConDirMatchingFilter`) and the Source Matching Filter (`orclODIPOIDMatchingFilter`) fields, enter the appropriate values for the `searchfilter` attribute. Instructions for specifying the `searchfilter` attribute are provided in the section "Filtering Changes with an LDAP Search" on page 6-19.
8. Choose **OK**.

To customize the search filter by using the manageSyncProfiles command:

1. Enter the following command to customize the Connected Directory Matching Filter (`orclODIPConDirMatchingFilter`) attribute:

```
manageSyncProfiles update -h host -p port -D WLS_login_ID
-pf synchronization_profile_name -params "odip.profile.condirfilter
searchfilter=(|(objectclass=group)(objectclass=organizationalunit)(&(objectclas
s=user)(!(objectclass=computer))))"
```

2. Enter the following command to customize the OID Matching Filter (`orclODIPOIDMatchingFilter`) attribute:

```
manageSyncProfiles update -h host -p port -D WLS_login_ID
-pf synchronization_profile_name -params "odip.profile.oidfilter
orclObjectGUID"
```

Note: All attributes specified in the `searchfilter` attribute should be configured as indexed attributes in Microsoft Active Directory.

See Also: The appendix about the LDAP filter definition in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for instructions on configuring an LDAP search filter

18.3.4 Step 4: Customizing the ACLs

Customize ACLs as described in "Customizing Access Control Lists" on page 17-9.

18.3.5 Step 5: Customizing Attribute Mappings

When integrating with Microsoft Active Directory, the following attribute-level mapping is mandatory for all objects:

```
ObjectGUID:  :  : :orclObjectGUID:
ObjectSID:  :  : :orclObjectSID:
```

Example 18–1 Attribute-Level Mapping for the User Object in Microsoft Active Directory

```
SAMAccountName:1: :user:orclADSAMAccountName: :orclADUser
userPrincipalName: : :user:orclADUserPrincipalName::orclADUser:userPrincipalName
```

Example 18–2 Attribute-Level Mapping for the Group Object in Microsoft Active Directory

```
SAMAccountName:1: :group:orclADSAMAccountName: :orclADGroup
```

In the preceding examples, `SAMAccountName` and `userPrincipalName` from Microsoft Active Directory are mapped to `orclADSAMAccountName` and `orclADUserPrincipalName` in Oracle Internet Directory.

Customize the attribute mappings by following the instructions in "Customizing Mapping Rules" on page 17-12.

18.3.6 Step 6: Synchronizing with Multiple Microsoft Active Directory Domains

When synchronizing with multiple Microsoft Active Directory domains, you need separate import and export synchronization profiles for each domain in most cases. However, the profiles for each domain should be very similar. The only exception involves using Global Catalog with import synchronization profiles. In this case, you only need to create a single import synchronization profile for the entire Microsoft Active Directory forest. For more information, see "Configuration Required for Importing from Microsoft Active Directory to the Oracle Back-end Directory" on page 16-27.

> **Note:** Be sure to perform attribute and DN mapping before attempting to synchronize with multiple domains.

The best approach to creating separate import and export synchronization profiles for multiple domains is as follows:

1. Customize the import and export synchronization profiles for a single domain, using the procedures described earlier in this section.
2. Once you have finished customizing the import and export synchronization profiles for the first domain, use the `copy` operation of the `manageSyncProfiles` command to duplicate profiles, as follows:

   ```
   manageSyncProfiles copy -h host -p port -D WLS_login_ID
   -pf Original_Profile_Name -newpf New_Profile_Name
   ```

3. Use the `update` operation of the manageSyncProfiles command to customize the profiles for each additional Microsoft Active Directory domain, as follows:

   ```
   manageSyncProfiles update -h host -p port -D WLS_login_ID
   -pf Profile_Name -params "prop1 val1 prop2 val2 ..."
   ```

4. If necessary, update the connection details for each domain by following the instructions listed in "Configuring Connection Details" on page 6-2.
5. Update the last change number in the import and export synchronization profiles for each domain by running the following command:

```
manageSyncProfiles updatechgnum -h host -p port -D WLS_login_ID
-pf Profile_Name
```

6. Repeat Steps 2 through 5 for each Microsoft Active Directory domain with which you need to synchronize.

18.3.7 Step 7: Synchronizing Deletions from Microsoft Active Directory

To synchronize deletions in Microsoft Active Directory with the Oracle back-end directory, you must grant the necessary privilege to the Microsoft Active Directory user account that the Oracle directory integration server uses to perform synchronizations with Microsoft Active Directory. Microsoft Active Directory deletions can be synchronized with the Oracle back-end directory by querying for them in Microsoft Active Directory. The way to do this depends on whether you are using the DirSync approach or the USN-Changed approach.

For the DirSync approach, the Microsoft Active Directory user account that the Oracle Directory Integration Platform uses to access Microsoft Active Directory must have Domain Administrative permissions, belong to the Domain Administrators group, or be explicitly granted Replicating Directory Changes permissions.

> **See Also:** Article ID 303972 at `http://support.microsoft.com` for information on how to grant Replicating Directory Changes permissions

For the USN-Changed approach, the Microsoft Active Directory user account that the Oracle Directory Integration Platform uses to access Microsoft Active Directory must have "List Content" and "Read Properties" permission to the `cn=Deleted Objects` container of a given domain. In order to set these permissions, you must use the dsacls.exe command that is available with recent versions of Microsoft Active Directory Lightweight Directory Service (AD LDS), which was previously known as Active Directory Application Mode or ADAM.

Regardless of whether you are using the DirSync approach or the USN-Changed approach to synchronize deletions in Microsoft Active Directory with the Oracle back-end directory, if you create a matching filter for the `ActiveImport` profile (for the DirSync approach) or the `ActiveChgImp` profile (for the USN-Changed profile) be sure to include only the following key Microsoft Active Directory attributes:

- `ObjectGUID`
- `ObjectSID`
- `ObjectDistName`
- `USNChanged`

In you specify any attributes in a matching filter other than the preceding key attributes, deletions in Microsoft Active Directory are not propagated to the Oracle back-end directory.

See Also:

- Article ID 230113 at `http://support.microsoft.com` for more information on deleting items from Microsoft Active Directory
- The attribute reference chapter in *Oracle Fusion Middleware Reference for Oracle Identity Management* for a listing of the standard LDAP attributes that the Oracle back-end directory supports

18.3.8 Step 8: Synchronizing in SSL Mode

Configure the Microsoft Active Directory connector for synchronization in SSL mode by following the instructions in "Configuring the Connected Directory Connector for Synchronization in SSL Mode" on page 17-13.

18.3.9 Step 9: Synchronizing Passwords

To synchronize password changes from the Oracle back-end directory to Microsoft Active Directory, follow these steps:

Note: Password synchronization is not supported from an Oracle Unified Directory back-end directory or an Oracle Directory Server Enterprise Edition back-end directory to Microsoft Active Directory. Password synchronization is supported from an Oracle Internet Directory back-end directory to Microsoft Active Directory.

1. Configure the Oracle back-end directory, Oracle Directory Integration Platform, and Microsoft Active Directory to run in SSL server authentication mode.
2. Enable password synchronization from the Oracle back-end directory to Microsoft Active Directory by following the instructions in "Enabling Password Synchronization from the Oracle Back-end Directory to a Connected Directory" on page 17-15.
3. Configure the Microsoft Active Directory connector to synchronize passwords by installing and configuring the Oracle Password Filter for Microsoft Active Directory, as described in Chapter 19, "Deploying the Oracle Password Filter for Microsoft Active Directory".

18.3.10 Step 10: Configuring the Microsoft Active Directory External Authentication Plug-in

Configure the Microsoft Active Directory external authentication plug-in by following the instructions in "Configuring External Authentication Plug-ins" on page 17-16.

18.3.11 Step 11: Performing Post-Configuration and Administrative Tasks

Read Chapter 23, "Managing Integration with a Connected Directory" for information on post-configuration and ongoing administration tasks.

18.4 Using DirSync Change Tracking for Import Operations

By default, the import synchronization profile created with `expressSyncSetup` uses the USN-Changed approach for tracking changes. If you want to use the DirSync change tracking approach, be sure to perform the steps in this section before beginning synchronization.

> **Note:** You may want to back up your current import synchronization profile before performing the following procedures. You can create a backup copy of a profile by using the `copy` operation of the manageSyncProfiles command.

To modify the import synchronization profile to use the DirSync change tracking approach:

1. You can use the activeimp.cfg.master file, located in your $*ORACLE_HOME*/ldap/odi/conf directory, to change the import synchronization profile from the USN-Changed approach to DirSync. Use the following command to update the profile:

   ```
   manageSyncProfiles update -h host -p port -D WLS_login_ID -pf Profile_Name
   -params "odip.profile.configfile $ORACLE_
   HOME/ldap/odi/conf/activeimp.cfg.master"
   ```

2. Update the last change number by running the following command:

   ```
   manageSyncProfiles updatechgnum -h host -p port -D WLS_login_ID
   -pf Profile_Name
   ```

18.5 Configuring Windows Native Authentication

This section describes the system requirements and tasks for configuring Windows Native Authentication. It contains these topics:

- What are the System Requirements for Windows Native Authentication?
- Avoiding HTTP-401 Errors and Repeat Login Challenges for External Users
- Configuring Windows Native Authentication with a Single Microsoft Active Directory Domain
- Configuring Windows Native Authentication with Multiple Microsoft Active Directory Domains or Forests
- Implementing Fallback Authentication
- Understanding the Possible Login Scenarios

18.5.1 What are the System Requirements for Windows Native Authentication?

Windows Native Authentication is intended for intranet Web applications. Your intranet deployment must include the following:

- Windows 2000 server with Microsoft Active Directory
- Kerberos service account established for OracleAS Single Sign-On Server
- Oracle Application Server 11*g* Release 1 (11.1.1) infrastructure installed

Note: Although the sample configurations in this section are for UNIX/Linux, Oracle Fusion Middleware can also be installed on Microsoft Windows.

- OracleAS Single Sign-On Server middle tier configured to use a Kerberos realm
- Synchronization of Microsoft Active Directory with the Oracle back-end directory
- The Oracle back-end directory configured to use the Windows external authentication plug-in

Note: Your back-end directorymust be Oracle Internet Directory to use the external authentication plug-in. If your back-end directory is either Oracle Unified Directory or Oracle Directory Server Enterprise Edition, the external authentication plug-in is not supported.

18.5.2 Avoiding HTTP-401 Errors and Repeat Login Challenges for External Users

If only one Single Sign-On (SSO) server is configured, you cannot avoid the HTTP-401 response from the SSO server that is configured for Windows Native Authentication (WNA) for a website that can be accessed both internally by users who are Windows authenticated and also externally by users who are not in a Windows domain. If you are planning to use Windows Native Authentication, consider using a configuration comprised of two SSO servers, each with different IP addresses, to avoid HTTP-401 errors being sent to external users' browsers and being presented with multiple login challenges.

See Also: Refer to Note 417620.1 in My Oracle Support (formerly MetaLink) for more information. You can access My Oracle Support at: `http://metalink.oracle.com/`

18.5.3 Configuring Windows Native Authentication with a Single Microsoft Active Directory Domain

To set up Windows Native Authentication, configure the Oracle back-end directory, the OracleAS Single Sign-On Server, and the user's browser by performing the following tasks in the order listed.

Task 1: Configure the OracleAS Single Sign-On Server

To configure the single sign-on server, complete the tasks described in these topics:

- Set Up a Kerberos Service Account for the OracleAS Single Sign-On Server
- Update the krb5.conf File
- Run the OracleAS Single Sign-On Server Configuration Assistant on each Oracle Application Server Single Sign-On Host

Set Up a Kerberos Service Account for the OracleAS Single Sign-On Server

Create a service account for the OracleAS Single Sign-On Server in Microsoft Active Directory, then create a keytab file for the server, and map the service principal (the server) to the account name. The keytab file stores the server's secret key. This file enables the server to authenticate to the KDC. The service principal is the entity, in this case, the single sign-on server, to which the KDC grants session tickets.

1. Synchronize system clocks. The OracleAS Single Sign-On Server middle tier and the Windows 2000 server must match. If you omit this step, then authentication fails because there is a difference in the system time.Be sure the time, the date, and the time zones are synchronized.
2. Check the port number of the Kerberos server on the Microsoft Active Directory host. The port where the Kerberos server listens is selected from `/etc/services` by default. On Windows systems, the services file is found at *system_drive*:\WINNT\system32\drivers\etc. The service name is Kerberos. Typically the port is set to `88/udp` and `88/tcp` on the Windows 2000 server. When added correctly to the services file, the entries for these port numbers are:

   ```
   kerberos5       88/udp          kdc             # Kerberos key server
   kerberos5       88/tcp          kdc             # Kerberos key server
   ```

3. In the hosts file located in the same directory as the services file, check the entry for the single sign-on middle tier. The fully qualified host name, which refers to the physical host name of the Oracle Application Server Single Sign-On server, must appear after the IP address and before the short name. The following is an example of a correct entry:

   ```
   130.111.111.111 sso.MyCompany.com sso loghost
   ```

4. Perform the following tasks to create a user account and keytab file in Microsoft Active Directory that will be used by the logical Oracle Application Server Single Sign-On host:

   a. Log in to the Microsoft Active Directory Management tool on the Windows 2000 server, then choose **Users**, then **New**, then **user**.

      Enter the name of the OracleAS Single Sign-On Server host, omitting the domain name. For example, if the host name is `sso.MyCompany.com`, then enter `sso`. This is the account name in Microsoft Active Directory.

      Note the password that you assigned to the account. You will need it later. Do *not* select **User must change password at next logon**.

   b. Create a keytab file for the OracleAS Single Sign-On Server, and map the account name to the service principal name.You perform both tasks by running the following command on the Windows 2000 server:

      ```
      C:> Ktpass -princ HTTP/sso.MyCompany.com@MyCompany.com -pass password
      -mapuser sso -out sso.keytab
      ```

      The `-princ` argument is the service principal. Specify the value for this argument by using the format `HTTP/`*`single_sign-on_host_name`*`@`*`KERBEROS_REALM_NAME`*. Note that `HTTP` and the Kerberos realm must be uppercase.

      Note that *`single_sign-on_host_name`* can be either the OracleAS Single Sign-On Server host itself or the name of a load balancer where multiple OracleAS Single Sign-On Server middle tiers are deployed. `MyCompany.com` is a fictitious Kerberos realm in Microsoft Active Directory. The user container is located within this realm. The `-pass` argument is the account password, the `-mapuser` argument is the account name of the OracleAS Single Sign-On Server middle tier, and the `-out` argument is the output file that stores the service key.

      Be sure to replace the example values given with values suitable for your installation. These values appear in boldface in the example.

Note:

- If the Ktpass is not found on your computer, then download the Windows Resource Kit from Microsoft to obtain the utility.
- The default encryption type for Microsoft Kerberos tickets is RC4-HMAC. Microsoft also supports DES-CBC and DES-CBC-MD5, two DES variants used in MIT-compliant implementations. Ktpass converts the key type of the KDC account from RC4_HMAC to DES.

5. For each Oracle Application Server Single Sign-On host, copy or FTP the keytab file, sso.keytab to the OracleAS Single Sign-On Server middle tier, placing it in $*ORACLE_HOME*/j2ee/OC4J_SECURITY/config. If you use FTP, be sure to transfer the file in binary mode.

   Be sure to give the Web server a unique identifier (UID) on the OracleAS Single Sign-On Server middle tier and to grant read permission for the file.

Update the krb5.conf File

You must update the krb5.conf file (krb5.ini on Windows) with the following information. If you do not update the krb5.conf file with the following information, the `kinit` test of the newly generated keytab file will fail, and the keytab file will fail when used for Windows Native Authentication in OracleAS Single Sign-On Server.

Update the krb5.conf file with the following information:

- The default realm of the Active Directory, for example: `AD.UK.ORACLE.COM`
- The hostname of the server where Active Directory resides, for example: `active.uk.oracle.com`
- The hostname of the server where OracleAS Single Sign-On Server resides, for example: `sso.uk.oracle.com`

For example, replace the ***marked-up text*** in the following text with the relevant default realm and KDC hostname, that is, the server where Active Directory resides:

Note: The krb5.conf file is case sensitive.

```
[libdefaults]
        default_realm = AD.UK.ORACLE.COM
        clockskew = 300
[realms]
        AD.UK.ORACLE.COM = {
                kdc = active.uk.oracle.com
        }
[domain_realm]
       .uk.oracle.com = AD.UK.ORACLE.COM
```

Run the OracleAS Single Sign-On Server Configuration Assistant on each Oracle Application Server Single Sign-On Host Running the ossoca.jar tool at this point does the following:

- Configures the Oracle Application Server Single Sign-On server to use the Sun JAAS login module

- Configures the server as a secured application

To run the ossoca.jar tool on the OracleAS Single Sign-On Server middle tier:

1. Back up the following configuration files:
   - $*ORACLE_HOME*/sso/conf/policy.properties
   - $*ORACLE_HOME*/j2ee/OC4J_SECURITY/config/jazn.xml
   - $*ORACLE_HOME*/opmn/conf/opmn.xml
   - $*ORACLE_HOME*/j2ee/OC4J_SECURITY/config/jazn-data.xml
   - $*ORACLE_HOME*/j2ee/OC4J_SECURITY/applications/sso/web/WEB-INF/web.xml
   - $*ORACLE_HOME*/j2ee/OC4J_SECURITY/application-deployments/sso/orion-application.xml
2. Run the ossoca.jar tool:
   - UNIX/Linux:

     ```
     $ORACLE_HOME/sso/bin/ssoca
     wna -mode sso
     -oh $ORACLE_HOME
     -ad_realm AD_REALM
     -kdc_host_port kerberos_server_host:port
     -verbose
     ```

   - Windows:

     ```
     %ORACLE_HOME%\jdk\bin\java -jar %ORACLE_HOME%\sso\lib\ossoca.jar
     wna -mode sso
     -oh %ORACLE_HOME%
     -ad_realm AD_REALM
     -kdc_host_port kerberos_server_host:port
     -verbose
     ```

   AD_REALM is the Kerberos realm in Microsoft Active Directory. This is the user container. Note from the syntax that this value must be entered in uppercase. The default port number for the KDC is usually 88. To confirm this, see step 2 in the section Set Up a Kerberos Service Account for the OracleAS Single Sign-On Server on page 18-9.
3. Step 2 shuts down the OracleAS Single Sign-On Server. Restart it:

   ```
   $ORACLE_HOME/opmn/bin/opmnctl startall
   ```

Task 2: Configure Internet Explorer for Windows Native Authentication

Configure Internet Explorer to use Windows Native Authentication. How you do this depends on which version you have.

- Internet Explorer 5.0 and Later
- Internet Explorer 6.0 Only

Internet Explorer 5.0 and Later

To configure Internet Explorer 5.0 and later, perform the following steps:

1. From the menu bar, select **Tools**, then, from the **Tools** menu, select **Internet Options**.
2. In the Internet Options dialog box, select the **Security** tab.

3. On the Security tab page, select **Local Intranet**, then select **Sites**.
4. In the Local intranet dialog box, select **Include all sites that bypass the proxy** server; then click **Advanced**.
5. In the advanced version of the Local intranet dialog box, enter the URL of the OracleAS Single Sign-On Server middle tier. For example:

   ```
   http://sso.mydomain.com
   ```

6. Click **OK** to exit the Local intranet dialog boxes.
7. In the Internet Options dialog box, select the **Security** tab; then choose **Local intranet**; then choose **Custom Level**.
8. In the Security Settings dialog box, scroll down to the User Authentication section and then select **Automatic logon only in Intranet zone**.
9. Click **OK** to exit the Security Settings dialog box.
10. From the menu bar, select **Tools**, then, from the **Tools** menu, select **Internet Options**.
11. In the Internet Options dialog box, select the **Connections** tab.
12. On the **Connections** tab page, choose **LAN Settings**.
13. Confirm that the correct address and port number for the proxy server are entered, then choose **Advanced**.
14. In the Proxy Settings dialog box, in the **Exceptions** section, enter the domain name for the OracleAS Single Sign-On Server (`MyCompany.com` in the example).
15. Click **OK** to exit the Proxy Settings dialog box.

Internet Explorer 6.0 Only

If you are using Internet Explorer 6.0, perform steps 1 through 12 in "Internet Explorer 5.0 and Later"; then perform the following steps:

1. From the menu bar, select **Tools**, then, from the **Tools** menu, select **Internet Options**.
2. In the Internet Options dialog box, select the **Advanced** tab.
3. On the **Advanced** tab page, scroll down to the Security section.
4. Select **Enable Integrated Windows Authentication (requires restart)**.

Task 3: Reconfigure Local Accounts

After configuring Windows Native Authentication, you must reconfigure accounts for the Oracle back-end directory administrator (`orcladmin` for Oracle Internet Directory) and other local Windows users whose accounts are in the Oracle back-end directory. If you omit this task, then these users will not be able to log in.

Use the Oracle Directory Services Manager interface for Oracle Internet Directory to perform these steps:

> **See:** The *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about using Oracle Directory Services Manager to configure Oracle Internet Directory.

1. Add the `orclADUser` class to the local user entry in the Oracle back-end directory.
2. Add the login ID of the local user to the `orclSAMAccountName` attribute in the user's entry. For example, the login ID of the orcladmin account is `orcladmin`.
3. Add the local user to the `exceptionEntry` property of the external authentication plug-in.

18.5.4 Configuring Windows Native Authentication with Multiple Microsoft Active Directory Domains or Forests

This section describes how to configure Windows Native Authentication with multiple Microsoft Active Directory domains or forests in the following types of deployments:

- Parent-child Microsoft Active Directory domains
- Microsoft Active Directory domains in the same forest with an established tree-root trust type
- Domains in different forests with an established forest trust type

> **Note:** Forest trust types are only supported in Windows Server 2003 and later versions of Windows operating systems.

To configure Windows Native Authentication with multiple Microsoft Active Directory domains or forests, perform the following tasks in the order listed:

Task 1: Verify that Trust is Established Between the Microsoft Active Directory Domains

Refer to your Microsoft Active Directory documentation for information on how to verify trust between multiple Microsoft Active Directory domains.

Task 2: Enable Windows Native Authentication with Oracle Application Server Single Sign-On through a Load Balancer or Reverse Proxy

Configure the Oracle Application Server Single Sign-On server to run behind a load balance or through reverse proxy by following the instructions in the advanced deployment options chapter of the *Oracle Enterprise Single Sign-On Suite Plus Administrator's Guide*

Task 3: Configure the OracleAS Single Sign-On Server

Configure each Oracle Application Server Single Sign-On server by following the instructions in "Task 1: Configure the OracleAS Single Sign-On Server" on page 18-9. Be sure to use the same Microsoft Active Directory realm and corresponding key distribution center (KDC) when configuring each physical Oracle Application Server Single Sign-On server instance. Also, be sure to use the load balance or reverse proxy name as the logical Oracle Application Server Single Sign-On host name.

Note: With multiple Microsoft Active Directory forests, the Oracle Application Server Single Sign-On server's logical host name must belong to one of the Microsoft Active Directory domains. For example, assume you have two Microsoft Active Directory forests and each forest contains a single domain. The domain in the first forest is named `engineering.mycompany.com` and the domain in the second forest is named `finance.mycompany.com`. The Oracle Application Server Single Sign-On server's logical host name must reside in either the `engineering.mycompany.com` or the finance.`mycompany.com domain`.

Task 4: Configure Internet Explorer for Windows Native Authentication

Configure the Oracle Application Server Single Sign-On server by following the instructions in "Task 2: Configure Internet Explorer for Windows Native Authentication" on page 18-12.

18.5.5 Implementing Fallback Authentication

The only browsers that support SPNEGO-Kerberos authentication are Internet Explorer 5.0 or later. OracleAS Single Sign-On Server provides fallback authentication support for unsupported browsers such as Netscape Communicator. Depending upon the type of browser and how it is configured, the user is presented with the OracleAS Single Sign-On Server login form or the HTTP basic authentication dialog box. In either case, the user must provide a user name and password. The user name consists of the Kerberos realm name and the user ID. The default way to enter the user name is shown in the following example.

```
domain_name\user_id
```

The following example, based on the example provided in "Set Up a Kerberos Service Account for the OracleAS Single Sign-On Server" on page 18-9, illustrates how to enter the user name.

```
MyCompany.COM\jdoe
```

Note that the user name and password are case sensitive. Additionally, password policies for Microsoft Active Directory do not apply. You can configure a different synchronization profile by using the Oracle Directory Integration Platform. If you do, the login format just provided does not apply.

Fallback authentication is performed against Microsoft Active Directory, using an external authentication plug-in for the Oracle back-end directory.

Note:

- HTTP basic authentication does not support logout. To clear credentials from the browser cache, users must close all open browser windows. Alternatively, they can log out of the Windows computer.
- In cases where basic authentication is invoked, users must set their language preference manually in Internet Explorer. Select **Internet Options** from the Tools menu, select **Languages**, and then enter the desired language.

18.5.6 Understanding the Possible Login Scenarios

Users may encounter a number of different login behaviors within Internet Explorer depending upon which version they are using. Table 18–1 on page 18-16 shows under what circumstances automatic sign-on and fallback authentication are invoked.

Table 18–1 Single Sign-On Login Options in Internet Explorer

Browser Version	Desktop Platform	Desktop Authentication Type	Integrated Authentication in Internet Explorer Browser	OracleAS Single Sign-On Server Login Type
5.0.1 or later	Windows 2000/XP	Kerberos V5	On	Automatic sign-on
5.0.1 or later but earlier than 6.0	Windows 2000/XP	Kerberos V5	Off	Single sign-on
6.0 or later	Windows 2000/XP	Kerberos V5 or NTLM	Off	HTTP basic authentication
5.0.1 or later but earlier than 6.0	Windows NT/2000/XP	NTLM	On or off	Single sign-on
6.0 or later	NT/2000/XP	NTLM	On	Single sign-on
5.0.1 or later	Windows 95, ME, Windows NT 4.0	Not applicable	Not applicable	Single sign-on
Earlier than 5.0.1	N/A	Not applicable	Not applicable	Single sign-on
All other browsers	All other platforms	Not applicable	Not applicable	Single sign-on

18.6 Configuring Synchronization of Oracle Internet Directory Foreign Security Principal References with Microsoft Active Directory

This section explains how to synchronize Oracle Internet Directory foreign security principal references with Microsoft Active Directory.

Although Microsoft Active Directory stores information for group members in a trusted domain as foreign security principal references, Oracle Internet Directory stores the DNs of these members as they appear in Oracle Internet Directory. This results in a mismatch between an entry and its value as a member of a group. The relationship between a user and a group cannot be directly established in Oracle Internet Directory.

To establish the relationship between users and groups, the member DNs that refer to the foreign security principals must be replaced by the DNs of the entries during the synchronization of such groups. This is called resolving foreign key references.

Note: Synchronization of foreign security principal references is supported only on Windows 2003.

Example 18–3 How Foreign Key References Are Resolved

The example in this section illustrates how foreign key references are resolved.

Assume that there are three domains: A, B and C.

Domain A has a one-way non-transitive trust to Domain B. It can have foreign security principal references for users and groups from Domain B.

Domain A has a one-way non-transitive trust to Domain C. It can have foreign security principal references for users and groups from Domain C.

Domain B has a one-way non-transitive trust to Domain C. It can have foreign security principal references for users and groups from Domain C.

In this example, the one-way non-transitive trusts are from Domain A to Domain B, from Domain A to Domain C, and from Domain B to Domain C.

Tasks to Resolve Foreign Key References

This section explains the steps for resolving foreign key references.

Task 1: Update Agent Configuration Information For each profile that can have foreign security principal references, perform the following steps. The sample configuration files are in the $*ORACLE_HOME*/ldap/odi/conf/ directory.

1. Copy the activeimp.cfg.fsp file. The following is an example of the activeimp.cfg.fsp file:

```
[INTERFACEDETAILS]
   Package: gsi
   Reader: ActiveReader
[TRUSTEDPROFILES]
   prof1 : <Name of the profile1>
   prof2 : <Name of the profile2>
[FSPMAXSIZE]
   val=10000
```

   The preceding example assumes you are using the DirSync change tracking approach. If you are using the USN-Changed approach for tracking changes, assign a value of `ActiveChgReader` to the Reader parameter.

2. In the activeimp.cfg.fsp file, under the `[TRUSTEDPROFILES]` tag, specify the profile names of the other domains that have foreign security principal references in this domain.

   Referring to Example 18–3 on page 18-16, agent configuration information for Domain A contains the following:

```
[INTERFACEDETAILS]
   Package: gsi
   Reader: ActiveReader
[TRUSTEDPROFILES]
   prof1: profile_name_for_domain_B
   prof2: profile_name_for_domain_C
```

   Agent configuration information for domain B contains the following:

```
[INTERFACEDETAILS]
   Package: gsi
   Reader: ActiveReader
[TRUSTEDPROFILES]
   prof1: profile_name_for_domain_C
```

   Agent configuration information for domain C has no changes because domain C has no foreign key references.

3. Under the `[FSPMAXSIZE]` tag, specify the foreign security principal cache size. This can be the average number of foreign security principals you can have. A sample value of `1000` is specified in the activeimp.cfg.fsp file.

4. Load the new agent configuration information file by using the update operation of the manageSyncProfiles command as follows:

```
manageSyncProfiles update -h host -p port -D WLS_login_ID
-pf profile_name_for_domain_A_or_B -params "odip.profile.configfile
activeimp.cfg.fsp"
```

5. Repeat this task for every profile of interest.

Task 2: Modify the Input Data Before Bootstrapping to Resolve the Foreign Security Principal References To do this, perform the following steps:

1. Get the LDIF dump from the Microsoft Active Directory with appropriate filtering so that the resultant LDIF file contains only the required objects, for example users and groups.

> **Note:** The command to dump entries from Microsoft Active Directory to Oracle Internet Directory is `ldifde`. This command can be run only from a Microsoft Windows environment.

2. Resolve the foreign security principal references by entering the following command:

```
$ORACLE_HOME/ldap/odi/admin/fsptodn
host=oid_host
port=oid_port
dn= OID_privileged_DN (that is, superuser or dipadmin user)
pwd=OID_password
profile=profile_name_for_domain_A_or_B
infile=input_filenameo_of_the_LDIF_dump_from_Active_Directory
outfile=output_filename
[sslauth=0|1]
```

   By default, `host` is set to *`local_host`*, `port` is set to `3060`, and `sslauth` is set to `0`.

> **Note:** You can verify the successful execution of the command by verifying that the output file contains no references to `cn=foreignsecurityprincipals` in the member attribute. This command performs no attribute-level mapping other than resolving foreign security principal references.

3. Use the syncProfileBootstrap command to bootstrap the data from Microsoft Active Directory to Oracle Internet Directory.

> **See Also:** "Bootstrapping Data Between Directories" on page 23-2

Task 3: Update the Mapping Rules to Resolve the Foreign Security Principals During Synchronization After bootstrapping, modifications to groups must be reflected in Oracle Internet Directory with the correct group membership values. The `fsptodn` mapping rule enables you to do this when you synchronize. Modify this mapping rule in every profile that needs foreign security principal resolution. Referring to Example 18–3 on page 18-16, the mapping rules must be modified for Domains A and B.

If you do not have DN mapping, then change your mapping rule for the `member` attribute to the following:

```
member: : :group:uniquemember: :groupofUniqueNames: fsptodn(member)
```

If you have DN mapping, then change the mapping rules as follows:

1. Add the DN mapping rules corresponding to each of the trusted domains. This is used to resolve the correct domain mapping. Referring to Example 18–3 on page 18-16, the `domainrules` in the mapping file for Domain A should have content similar to the following:

   ```
   DOMAINRULES
   <Src Domain A >:<Dst domain A1 in OID>
   <Src Domain B >:< Dst domain B1 in OID>
   <Src Domain C>:<Dst domain C1 in OID>
   ```

2. Change your mapping rule for the `member` attribute to:

   ```
   member:::group:uniquemember::groupofUniqueNames:dnconvert(fsptodn(member))
   ```

3. Upload the mapping file for the different profiles using the `update` operation of the manageSyncProfiles command, as follows:

   ```
   manageSyncProfiles update -h host -p port -D WLS_login_ID
   -pf Profile_Name -file File_Name
   ```

18.7 Switching to a Different Microsoft Active Directory Domain Controller in the Same Domain

This section explains how to change the Microsoft Active Directory domain controller to which changes are exported. There are two methods, one for the USN-Changed approach and the other for the DirSync approach.

How to Change the Microsoft Active Directory Domain Controller by Using the USN-Changed Approach

If you are using the USN-Changed approach, then perform the following:

1. Disable the current running profile. Modify the Microsoft Active Directory host connection information, that is, host, port, user, password, to point to the new host. Usually, the host name is the only item that you need to update.
2. Obtain the current value of the `highestCommittedUSN` by searching the new domain controller's root DSE for the current highest `USNChanged` value (attribute value of the `highestCommittedUSN` attribute of the root DSE):

   ```
   ldapsearch -h host -p port -b "" -s base -D binddn -q \
   "objectclass=*" highestCommittedUSN
   ```

   Note: You will be prompted for the password.

3. Use Oracle Directory Integration Platform to run a full synchronization from Microsoft Active Directory.

   a. Run `ldifde`, the command to dump entries from Microsoft Active Directory to the Oracle back-end directory, using the intended LDAP search scope and search filter. Normally, the search filter should be the same as that specified in

the running profile. For example, the following search filter is set in the sample properties file. Note that `ldifde` can be run only from a Microsoft Windows environment.

```
searchfilter=(&(|(objectclass=user)(objectclass=organizationalunit))(!(obje
ctclass=group)))
```

Essentially, run `ldifde` with a search scope and search filter that retrieves all Oracle back-end directory objects (entries) that were configured to be synchronized with Microsoft Active Directory by the running profile.

b. Run Oracle Directory Integration Platform to upload the LDIF file generated in Step a using the same profile.

4. After the full synchronization is completed, update the `lastchangenumber` attribute with the `highestCommittedUSN` value obtained in Step 2.
5. Resume the normal synchronization, that is, incremental synchronization from Microsoft Active Directory using `USNChanged` attribute.

How to Change the Microsoft Active Directory Domain Controller by Using the DirSync Approach

If you are using the DirSync approach, perform the following steps:

1. Stop the current profile that is running.
2. Use the `copy` operation of the manageSyncProfiles command to create a new profile exactly the same as the profile already being used. In the newly created profile, modify the Microsoft Active Directory host connection information, that is, host, port, user, password, to point to the new host. Usually, the host name is the only item you need to update.
3. Resume normal synchronization with the modified profile. Note that all the domain controllers must be in the same Microsoft Active Directory domain.

18.8 Configuring the Microsoft Active Directory Connector for Microsoft Active Directory Lightweight Directory Service

The Microsoft Active Directory connector can be used for synchronizing the entries between Microsoft Active Directory Lightweight Directory Service (AD LDS), which was previously known as Active Directory Application Mode or ADAM, and the Oracle back-end directory.

18.9 Configuring the Microsoft Active Directory Connector for Microsoft Exchange Server

The Microsoft Active Directory Connector can provision users in Microsoft Exchange in deployments that have Microsoft Active Directory Server 2000 or later as their identity store.

You can use either of the following methods to configure the Microsoft Active Directory connector for Microsoft Exchange Server:

- Use the Oracle Enterprise Manager Fusion Middleware Control user interface
- Use the `manageSyncProfiles` command at a command line

See the following sections for details.

To further customize your integration with Microsoft Exchange, follow the instructions in "Configuring Advanced Integration with Microsoft Active Directory" on page 18-2.

See Also: *Oracle Application Server MS Office Developer's Guide*

18.9.1 To Enable Microsoft Exchange User Synchronization From the User Interface

1. Use the Oracle Enterprise Manager Fusion Middleware Control to create a synchronization profile, as described in "Creating Synchronization Profiles" on page 7-1.

   On the **General** tab, set the **Use DIP-OID as?** field to **Source** and select **MS Exchange Server** from the **Type** list.

   On the **Mapping** tab, in addition to creating domain mapping rules, you need to create two attribute mapping rules. Following are instructions on how to create the mapping rules.

2. On the **Mapping** tab, click **Create** in the **Attribute Mapping Rules** section.

   The Add Attribute Mapping Rule dialog box opens.

3. Create the first (of two) attribute mapping rules using the following steps:

   a. Select `inetorgperson` from the **Source ObjectClass** drop-down menu.

   b. Select the **Single Attribute** option, then select `uid` from the **Source Attribute** drop-down menu.

   c. Select `User` from the **Destination ObjectClass** drop-down menu.

   d. Select `homeMTA` from the **Destination Attribute** drop-down menu.

   e. Type the value of the MTA DN in the **Mapping Expression** field.

      To obtain the value for `homeMTA`, run a simple LDAP search query on any user in Active Directory.

      The MTA DN follows this format:

      ```
      CN=Microsoft MTA,CN=<host>,CN=Servers,CN=First
      Administrative Group,CN=Administrative
      Groups,CN=Oracle,CN=Microsoft
      Exchange,CN=Services,CN=Configuration,<Domain_DN>
      ```

      For example:

      ```
      CN=Microsoft MTA,CN=DADVMN0152,CN=Servers,CN=First
      Administrative Group,CN=Administrative
      Groups,CN=Oracle,CN=Microsoft
      Exchange,CN=Services,CN=Configuration,DC=diptest,DC=us,DC=
      oracle,DC=com
      ```

   f. Click **OK** to save the rule.

4. Create the second attribute mapping rule using the following steps:

   a. Select `inetorgperson` from the **Source ObjectClass** drop-down menu.

   b. Select the **Single Attribute** option, then select `uid` from the **Source Attribute** drop-down menu.

   c. Select `User` from the **Destination ObjectClass** drop-down menu.

   d. Select `homeMDB` from the **Destination Attribute** drop-down menu.

   e. Type the value of the MDB DN in the **Mapping Expression** field.

To obtain the value for homeMDB, run a simple LDAP search query on any user in Active Directory.

The MDB DN follows this format:

```
CN=Mailbox Store (<host>),CN=First Storage Group,
CN=InformationStore,CN=<host>,CN=Servers,CN=First
Administrative Group,CN=Administrative
Groups,CN=Oracle,CN=Microsoft
Exchange,CN=Services,CN=Configuration,<Domain_DN>
```

For example:

```
CN=Mailbox Store (DADVMN0152),CN=First Storage Group,
CN=InformationStore,CN=DADVMN0152,CN=Servers,CN=First
Administrative Group,CN=Administrative
Groups,CN=Oracle,CN=Microsoft Exchange,CN=Services,
CN=Configuration,DC=diptest,DC=us,DC=oracle,DC=com
```

f. Click **OK** to save the rule.

18.9.2 To Enable Microsoft Exchange User Synchronization From the Command Line

1. Use the manageSyncProfiles command, as described in "Syntax for manageSyncProfiles" on page 7-16.

   When you run the command, specify ExchangeServer2003 as the value assigned to the -conDirType argument.

   Import and export profiles will be created. The import profile is based on the Active Directory USN template profile and the export profile is based on the Exchange Sever template profile.

2. Edit the msexchangeexp.map.master mapping file and create domain mapping rules and attribute mapping rules. Details about how to create the attribute mapping rules are included below. For general information about mapping rules, see "Customizing Mapping Rules" on page 17-12.

   a. Open the msexchangeexp.map.master mapping file (located in ORACLE_HOME/ldap/odi/conf/) and locate the following attribute mapping rule:

      ```
      uid:: :inetorgperson:homeMTA: :User:'%DN_OF_MTA%'
      ```

   b. Replace %DN_OF_MTA% with the actual value of the MTA DN.

      To obtain the value for homeMTA, run a simple LDAP search query on any user in Active Directory.

      The MTA DN follows this format:

      ```
      CN=Microsoft MTA,CN=<host>,CN=Servers,CN=First
      Administrative Group,CN=Administrative
      Groups,CN=Oracle,CN=Microsoft
      Exchange,CN=Services,CN=Configuration,<Domain_DN>
      ```

      For example:

      ```
      CN=Microsoft MTA,CN=DADVMN0152,CN=Servers,CN=First
      Administrative Group,CN=Administrative
      Groups,CN=Oracle,CN=Microsoft
      Exchange,CN=Services,CN=Configuration,DC=diptest,DC=us,DC=
      oracle,DC=com
      ```

c. In the `msexchangeexp.map.master` mapping file locate the following attribute mapping rule:

```
uid:: :inetorgperson:homeMDB: :User:'%DN_OF_MDB%'
```

d. Replace `%DN_OF_MDB%` with the actual value of the MDB DN.

To obtain the value for `homeMDB`, run a simple LDAP search query on any user in Active Directory.

The MDB DN follows this format:

```
CN=Mailbox Store (<host>),CN=First Storage Group,
CN=InformationStore,CN=<host>,CN=Servers,CN=First
Administrative Group,CN=Administrative
Groups,CN=Oracle,CN=Microsoft
Exchange,CN=Services,CN=Configuration,<Domain_DN>
```

For example:

```
CN=Mailbox Store (DADVMN0152),CN=First Storage Group,
CN=InformationStore,CN=DADVMN0152,CN=Servers,CN=First
Administrative Group,CN=Administrative
Groups,CN=Oracle,CN=Microsoft Exchange,CN=Services,
CN=Configuration,DC=diptest,DC=us,DC=oracle,DC=com
```

e. Save your changes.

3. Edit the `msexchangeexp.properties` file (located in `ORACLE_HOME/ldap/odi/conf/`) and specify the following:

- Microsoft Exchange server host name
- Microsoft Exchange server port number
- User name
- Password
- The location of the `msexchangeexp.map.master` file and the `activeexp.cfg.master` file

19

Deploying the Oracle Password Filter for Microsoft Active Directory

This chapter explains how to install and configure the Oracle Password Filter for Microsoft Active Directory.

> **Note:** to use the Oracle Password Filter for Microsoft Active Directory, your Oracle back-end directory must be Oracle Internet Directory. The Oracle Unified Directory back-end directory and the Oracle Directory Server Enterprise Edition back-end directory do not support integration with the Oracle Password Filter for Microsoft Active Directory.

This chapter contains the following sections:

For help troubleshooting an issue with the Oracle Password Filter for Microsoft Active Directory, see the following topic in Appendix E, "Troubleshooting the Oracle Directory Integration Platform."

> **Note:** The installation file for the Oracle Password Filter for Microsoft Active Directory is located on the Oracle Application Server CD-ROM (Disk 1) for Windows.
>
> A 32-bit version and a 64-bit version of the password filter application are provided. The 32-bit version should only be installed on a 32-bit OS, and the 64-bit version should only be installed on a 64-bit OS. See "Installing the Oracle Password Filter for Microsoft Active Directory" on page 19-8 for installation instructions.

19.1 Overview of the Oracle Password Filter for Microsoft Active Directory

This section describes the purpose of the Oracle Password Filter for Microsoft Active Directory and how it works. It contains these topics:

- What is the Oracle Password Filter for Microsoft Active Directory?
- How Does the Oracle Password Filter for Microsoft Active Directory Work?
- How Do I Deploy the Oracle Password Filter for Microsoft Active Directory?

19.1.1 What is the Oracle Password Filter for Microsoft Active Directory?

Oracle Directory Integration Platform enables synchronization between the Oracle back-end directory (the back-end directory must be Oracle Internet Directory to use Oracle Password Filter for Microsoft Active Directory) and Microsoft Active Directory. The Oracle Directory Integration Platform can retrieve all Microsoft Active Directory attributes with the exception of user passwords. Oracle Application Server Single Sign-On uses an external authentication plug-in to verify user credentials in Microsoft Active Directory and automatically store the updated password in the Oracle back-end directory. Applications such as Oracle Database Enterprise User Security that do not use Oracle Application Server Single Sign-On can use the Oracle Password Filter for Microsoft Active Directory to retrieve passwords from Microsoft Active Directory and store the password in the Oracle back-end directory.

Note: Your Oracle back-end directory must be Oracle Internet Directory to support Enterprise User Security. The Oracle Unified Directory back-end directory and the Oracle Directory Server Enterprise Edition back-end directory do not support integration with other Fusion Middleware components, including Enterprise User Security.

When users change their passwords from their desktops, the updated password is automatically synchronized with the Oracle back-end directory. More specifically, the Oracle Password Filter for Microsoft Active Directory monitors Microsoft Active Directory for password changes, which it then stores in the Oracle back-end directory. This allows users to be authenticated with their Microsoft Active Directory credentials and authorized to access resources by using information stored in the Oracle back-end directory. Storing Microsoft Active Directory user credentials in the Oracle back-end directory also provides a high availability solution in the event that the Microsoft Active Directory server is down. The Oracle Password Filter is installed on each Microsoft Active Directory server and automatically forwards password changes to the Oracle back-end directory.

Note: Enterprise User Security can only verify user credentials that are stored in the Oracle Internet Directory back-end directory. For this reason, to verify user credentials in Microsoft Active Directory with Enterprise User Security, you must use the Oracle Password Filter to retrieve passwords from Microsoft Active Directory into the Oracle Internet Directory back-end directory.

The Oracle Password Filter for Microsoft Active Directory does not require the Oracle Directory Integration Platform to synchronize passwords from Microsoft Active

Directory to the Oracle back-end directory. The only requirement is that users synchronized from Microsoft Active Directory to the Oracle back-end directory must include the `ObjectGUID` attribute value to identify the user in both directories. The Oracle Password Filter for Microsoft Active Directory does not enforce password policies, or differences in password policies, between Microsoft Active Directory and the Oracle back-end directory. Instead, the system administrator must ensure that the password policies are consistent in both directories.

Password change requests occur when an account is created, an administrator resets a user's password, or when a user changes his or her own password. In order for the Oracle Password Filter for Microsoft Active Directory to capture Microsoft Active Directory passwords, one of these events must occur. Passwords that were set prior to installing the Oracle Password Filter for Microsoft Active Directory cannot be captured unless a system administrator forces a global password change request to all users.

Note: The Oracle Password Filter for Microsoft Active Directory only captures password changes for 32-bit or higher Windows systems that have been integrated with Microsoft Active Directory.

19.1.2 How Does the Oracle Password Filter for Microsoft Active Directory Work?

This section describes how the Oracle Password Filter for Microsoft Active Directory works. It contains these topics:

- How Clear Text Password Changes are Captured
- Password Changes are Stored when the Oracle Back-end Directory is Unavailable
- Password Synchronization is Delayed Until Microsoft Active Directory Users are Synchronized with Oracle Identity Management
- Password Bootstrapping

19.1.2.1 How Clear Text Password Changes are Captured

When a password change request is made, the Local Security Authority (LSA) of the Windows operating system calls the Oracle Password Filter for Microsoft Active Directory package that is registered on the system. When the LSA calls the Oracle Password Filter for Microsoft Active Directory package, it passes to it the user name and changed password. The Oracle Password Filter for Microsoft Active Directory then performs the synchronization.

19.1.2.2 Password Changes are Stored when the Oracle Back-end Directory is Unavailable

When the Oracle back-end directory is unavailable, the password change events are archived securely and the encrypted passwords are stored in the Microsoft Active Directory. The Oracle Password Filter for Microsoft Active Directory attempts to synchronize these entries until it reaches the specified maximum number of retries.

19.1.2.3 Password Synchronization is Delayed Until Microsoft Active Directory Users are Synchronized with Oracle Identity Management

The Oracle Password Filter for Microsoft Active Directory is notified immediately when a new user is created in Microsoft Active Directory. However, Oracle Directory Integration Platform will not synchronize entries until the next scheduled synchronization interval. For this reason, passwords for new user entries are stored in encrypted format in Microsoft Active Directory until the next synchronization. The

Oracle Password Filter for Microsoft Active Directory then attempts to synchronize these entries until it reaches the specified maximum number of retries.

19.1.2.4 Password Bootstrapping

Because the original clear text form of a password is not retrievable by the Oracle Password Filter for Microsoft Active Directory, you cannot perform initial bootstrapping to synchronize passwords from Microsoft Active Directory to the Oracle back-end directory. However, you can instruct users to change their passwords or force a password change for all users in Microsoft Active Directory by changing the password expiration policy.

19.1.3 How Do I Deploy the Oracle Password Filter for Microsoft Active Directory?

The general procedures for installing and configuring the Oracle Password Filter for Microsoft Active Directory are as follows;

1. Enable synchronization between the Oracle back-end directory (Oracle Internet Directory) and Microsoft Active Directory by following the instructions described in Chapter 18, "Integrating with Microsoft Active Directory".
2. Configure and test the Oracle back-end directory in SSL server authentication mode by following the instructions in "Configuring and Testing Oracle Internet Directory with SSL Server-Side Authentication" on page 19-4.
3. Import the Oracle back-end directory trusted server certificate into the Microsoft Active Directory domain controller by following the instructions in "Importing a Trusted Certificate into a Microsoft Active Directory Domain Controller" on page 19-5.
4. Verify that the Oracle back-end directory and Microsoft Active Directory can communicate with SSL server authentication by following the instructions in "Testing SSL Communication Between Oracle Internet Directory and Microsoft Active Directory" on page 19-6.
5. Install the Oracle Password Filter for Microsoft Active Directory by following the instructions in "Installing the Oracle Password Filter for Microsoft Active Directory" on page 19-8.
6. Configure the Oracle Password Filter for Microsoft Active Directory by following the instructions in "Reconfiguring the Oracle Password Filter for Microsoft Active Directory" on page 19-16.

19.2 Configuring and Testing Oracle Internet Directory with SSL Server-Side Authentication

The Oracle Password Filter communicates password changes from Microsoft Active Directory to Oracle Internet Directory using the Secure Socket Layer (SSL) protocol, which provides data encryption and message integrity for a TCP/IP connection. More specifically, to synchronize password changes between Oracle Internet Directory and Microsoft Active Directory, you must use SSL server authentication mode, which allows a client to confirm a server's identity.

When combined with digital certificates, SSL also provides both server authentication and client authentication. Server authentication with SSL requires that you install a digital certificate on the server side of the communications link. When an SSL transaction is initiated by a client, the server sends its digital certificate to the client. The client examines the certificate to validate that the server has properly identified

itself, including verifying that the certificate was issued by a trusted Certificate Authority (CA).

The subject attribute of the Oracle Internet Directory server certificate must match the Oracle Internet Directory server hostname. For example, if the Oracle Internet Directory server hostname is oid.oracle.com, then the subject attribute of the Oracle Internet Directory server certificate must also be oid.oracle.com. If the subject attribute of the Oracle Internet Directory server certificate **does not** match the Oracle Internet Directory server hostname, the Microsoft Active Directory password filter API will not accept the Oracle Internet Directory server certificate as being valid, despite the `ldapbind -U 2` command's success. Oracle Internet Directory configured for Server authentication is also referred to as SSL type 2.

In the case of Oracle Internet Directory and Microsoft Active Directory integration, Oracle Internet Directory is the server and Microsoft Active Directory is the client. The Oracle Password Filter for Microsoft Active Directory uses SSL to protect the password during transmission between the Microsoft Active Directory domain controller and the Oracle Internet Directory server.

> **Note:** The certificate you use with the Oracle Password Filter for Microsoft Active Directory can be generated by any X.509-compliant certificate authority capable of accepting PKCS#10 standard certificate requests and producing certificates compliant with the X.509, Version 3, ISO standard and with RFC 2459.

To configure and test Oracle Internet Directory with SSL server-side authentication, refer to *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

19.3 Importing a Trusted Certificate into a Microsoft Active Directory Domain Controller

Server-authenticated SSL communication between a Microsoft Active Directory domain controller and Oracle Internet Directory will fail if the domain controller does not recognize the Oracle Internet Directory SSL certificate as valid. In order for a domain controller to accept an Oracle Internet Directory SSL certificate, you must use the Microsoft Management Console to import the certificate authority's trusted certificate into the domain controller.

To use the Microsoft Management Console to import the certificate authority's trusted certificate into the domain controller:

1. Select **Run** from the Windows **Start** menu. The Run dialog box displays. In the Run dialog box, type **mmc**, and then click **OK**. The Microsoft Management Console window displays.
2. Select **Add/Remove Snap-in** from the **File** menu. The Add/Remove Snap-in dialog box displays.
3. In the Add/Remove Snap-in dialog box, click **Add**. The Add Standalone Snap-in dialog box displays.
4. In the Add Standalone Snap-in dialog box, select **Certificates**, and then click **Add**. The Certificates snap-in dialog box displays, prompting you to select an option for which the snap-in will manage certificates.
5. In the Certificates snap-in dialog box, select **Computer Account**, and then click **Next**. The Select Computer dialog box displays.

6. In the Select Computer dialog box, select **Local Computer**, and then click **Finish**.
7. Click **Close** in the Add Standalone Snap-in dialog box, and then click **OK** in the Add/Remove Snap-in dialog box. The new console displays *Certificates (Local Computer)* in the console tree.
8. In the console tree, expand **Certificates (Local Computer)**, and then click **Trusted Root Certification Authority**.
9. Point to **All Tasks** on the **Action** menu, and then select **Import**. The Welcome page of the Certificate Import Wizard displays. Click **Next** to display the File to Import page.
10. On the **File to Import** page, enter the path and file name of the certificate authority's trusted root certificate, or click **Browse** to search for a file, and then click **Next**. The Certificate Store page displays.
11. On the Certificate Store page, select **Place all certificates in the following store**. If Trusted Root Certification Authorities is not already selected as the certificate store, click **Browse** and select it. Click **Next**. The Completing the Certificate Import page displays.
12. On the Completing the Certificate Import page, click **Finish**. A dialog box displays indicating that the import was successful. Click **OK**.
13. Click **Save** from the **File** menu. The Save As dialog box displays. Enter a name for the new console, and then click **Save**.
14. Close **Microsoft Management Console**.

> **Note:** For help on importing a trusted certificate with Microsoft Management Console, refer to your Windows product documentation or visit Microsoft Help and Support at `http://support.microsoft.com`.

19.4 Testing SSL Communication Between Oracle Internet Directory and Microsoft Active Directory

The Oracle Password Filter for Microsoft Active Directory installs a command named `ldapbindssl` on the domain controller that you can use to test SSL communication between Oracle Internet Directory and Microsoft Active Directory.

> **Note:** The `ldapbindssl` binary is included in the Oracle Password Filter for Microsoft Active Directory installation. You cannot execute the `ldapbindssl` command without first installing the Oracle Password Filter for Microsoft Active Directory.

The syntax for the `ldapbindssl` is as follows:

```
ldapbindssl -h oid_hostname -p ssl_port -D binddn -w password
```

To test SSL connectivity from Microsoft Active Directory to Oracle Internet Directory:

1. Open a command prompt window on the domain controller and navigate to the folder where you installed the Oracle Password Filter for Microsoft Active Directory.

2. Enter the `ldapbindssl` command to test SSL communication with Oracle Internet Directory. For example, the following command attempts to bind to an Oracle Internet Directory host named oraas.mycompany.com on SSL port 3133:

   ```
   ldapbindssl -h oraas.mycompany.com -p 3133 -D binddn -w password
   ```

   If the `ldapbindssl` command is successful, the following response is returned:

   ```
   bind successful
   ```

   If the `ldapbindssl` command is not successful, the following response is returned:

   ```
   Cannot connect to the LDAP server
   ```

   If you cannot connect from Microsoft Active Directory to Oracle Internet Directory in SSL mode, verify that you successfully imported a trusted certificate into your Microsoft Active Directory domain controller, as described in "Importing a Trusted Certificate into a Microsoft Active Directory Domain Controller" on page 19-5.

3. Close the command prompt window.

19.5 Installing and Reconfiguring the Oracle Password Filter for Microsoft Active Directory

This section describes how to install and reconfigure the Oracle Password Filter for Microsoft Active Directory. It contains these topics:

- Installing the Oracle Password Filter for Microsoft Active Directory
- Reconfiguring the Oracle Password Filter for Microsoft Active Directory

Before you install or reconfigure the Oracle Password Filter for Microsoft Active Directory, be sure to collect the necessary configuration parameters for Microsoft Active Directory and for Oracle Internet Directory. Table 19–1 lists the configuration parameters you will need for Microsoft Active Directory and Table 19–2 lists the configuration parameters you will need for Oracle Internet Directory.

Table 19–1 Oracle Password Filter Configuration Parameters for Microsoft Active Directory

Parameter	Description
Domain	The Microsoft Active Directory domain for this domain controller. This value is typically the DNS domain name, in the form *mycompany.com*.
Base DN	The container in the Microsoft Active Directory DIT where the Oracle Password Filter searches for entries with changed passwords. If password propagation fails, the DNS of the failed password will be stored in an entry named `organizationalUnit` within the specified container. For this reason, the specified container should be capable of holding `organizationalUnit` objects. This value is typically in the form *dc=mycompany,dc=com*.
Port	The Microsoft Active Directory LDAP port (usually 3060).
Host	The IP address (NOT the host name) of the Microsoft Active Directory domain controller.
Microsoft Active Directory User	A user name with read privileges on the entire Microsoft Active Directory DIT and privileges to create an organizational unit and subtree entries under the Microsoft Active Directory base DN. Note that you must enter a user name and not the DN of an administrative user. This value is usually in the form *administrator@machine_name*.

Table 19–1 (Cont.) Oracle Password Filter Configuration Parameters for Microsoft Active Directory

Parameter	Description
Microsoft Active Directory User Password	The specified Microsoft Active Directory user's password.
Log File Path	A directory where log files will be written, such as E:\ADPasswordFilter\Log.

Table 19–2 Oracle Password Filter Configuration Parameters for Oracle Internet Directory

Parameter	Description
Base DN	The container in the Oracle Internet Directory DIT where the Oracle Password Filter searches for entries synchronized from Microsoft Active Directory. For example: `o=Microsoft Active Directory,c=us`.
Host	Specifies the host name where the Oracle Internet Directory LDAP processes are running. For Oracle Internet Directory installations running in a high availability configuration, use the virtual host name of the load balancer.
SSL Port	The Oracle Internet Directory port that is configured for SSL server authentication.
Non-SSL Port	The Oracle Internet Directory for unencrypted communication.
Oracle Internet Directory User	The distinguished name of an Oracle Internet Directory user with permissions to update user passwords in the base DN. For example: `cn=orcladmin`.
Oracle Internet Directory User Password	The specified Oracle Internet Directory user's password.

19.5.1 Installing the Oracle Password Filter for Microsoft Active Directory

This section describes how to install the Oracle Password Filter for Microsoft Active Directory on a domain controller.

> **Note:** The Microsoft Active Directory and Oracle Internet Directory configuration parameters listed in the following procedure are described in Table 19–1 and Table 19–2.

To install the Oracle Password Filter for Microsoft Active Directory on a domain controller:

1. Do the following:

   For 32-bit systems

   a. Locate the `setup.exe` file in the `utils\adpwdfilter` directory in the distribution package.

   b. Run the `setup.exe` command to extract the installation files to a directory on your domain controller.

   c. Navigate to the directory where you extracted the installation files and double-click `setup.exe`.

The Welcome page of the Oracle Password Filter for Microsoft Active Directory installation program displays, informing you that the program will install the Oracle Password Filter for Microsoft Active Directory.

For 64-bit systems

a. Updating the PATH environment variable on your Windows system is a prerequisite for installing the 64-bit version of the Oracle Password Filter for Microsoft Active Directory.

Append the following to the PATH environment variable on your Windows system:

```
C:\windows\SysWOW64
```

For instructions about how to edit Windows environment variables, refer to the following page:

http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sysdm_advancd_environmnt_addchange_variable.mspx

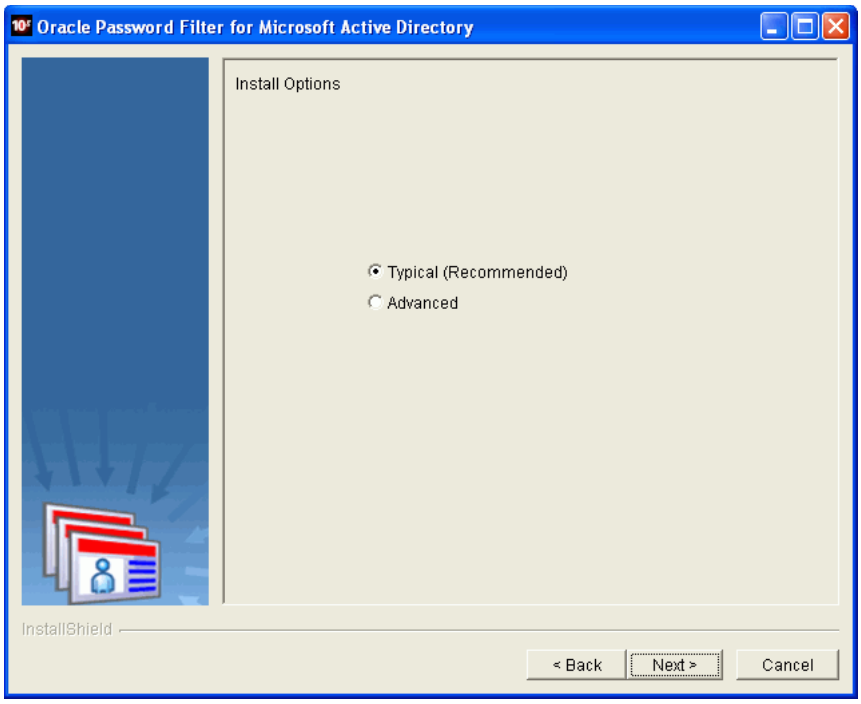
b. Locate the `setup.exe` file in the `utils\adpwdfilter\64bit` directory in the distribution package.

c. Run the `setup.exe` command to extract the installation files to a directory on your domain controller.

d. Navigate to the directory where you extracted the installation files and double-click `setup.exe`.

The Welcome page of the Oracle Password Filter for Microsoft Active Directory installation program displays, informing you that the program will install the Oracle Password Filter for Microsoft Active Directory.

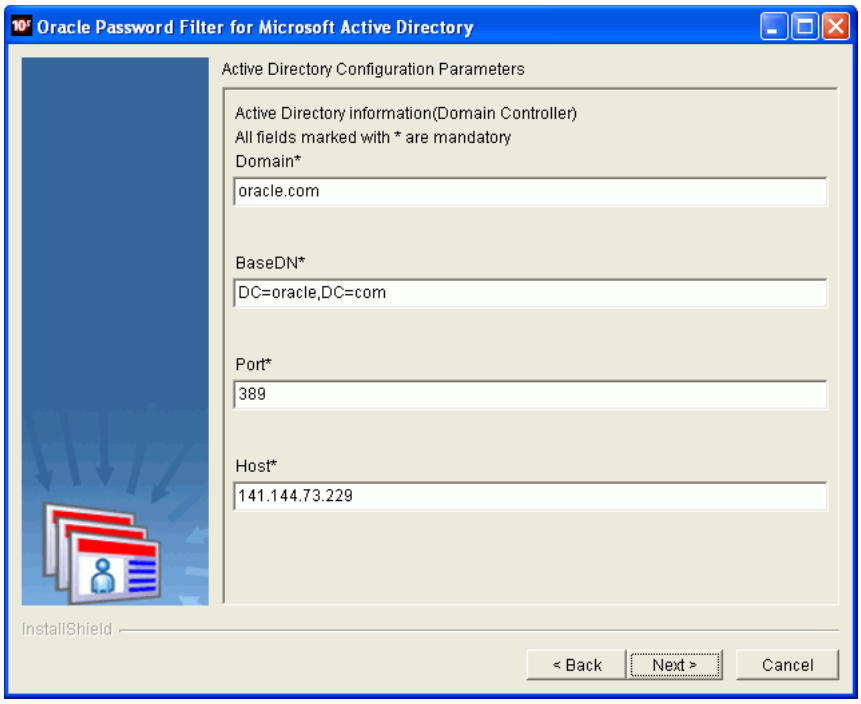
2. On the Welcome page, click **Next**. The Installation Requirements page displays, notifying you that SSL must be enabled between Oracle Internet Directory and Microsoft Active Directory and that installing the Oracle Password Filter for Microsoft Active Directory must restart your computer at the end of the installation process.
3. On the Installation Requirements page, click **Next**. The Installation Options page displays.



This image shows the Installation Options page of the Oracle Password Filter for Microsoft Active Directory installation program. This page contains two radio buttons, Typical (Recommended) and Advanced, along with Back, Next, and Cancel buttons. By default, the Typical (Recommended) radio button is selected.

****

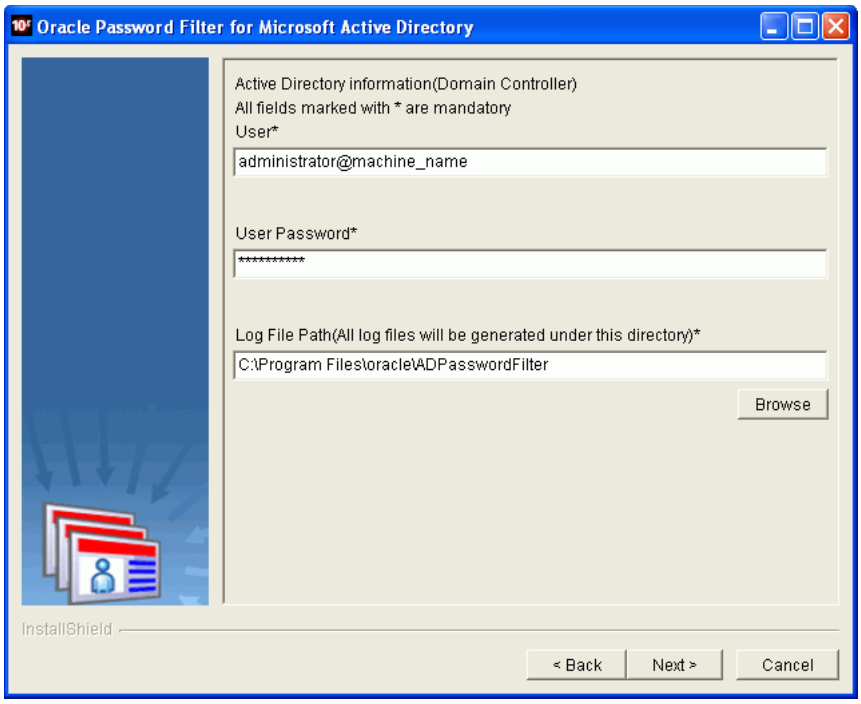
4. On the Installation Options page, select **Typical (Recommended)** or **Advanced**. Selecting to perform an advanced installation allows you to specify attributes for Oracle Internet Directory and Microsoft Active Directory later in the installation process (Step 13). Click **Next**. The Installation Location page displays, prompting you for the folder where you want to install Oracle Password Filter for Microsoft Active Directory.
5. On the Installation Location page, accept the default installation directory or enter a different directory. You can also select **Browse** to locate a different directory. Click **Next** after selecting an installation directory. The Microsoft Active Directory Configuration Parameters page displays.



This image shows the Microsoft Active Directory Configuration Parameters page of the Oracle Password Filter for Microsoft Active Directory installation program. This page contains Domain, Base DN, Port, and Host boxes, along with Back, Next, and Cancel buttons.

****

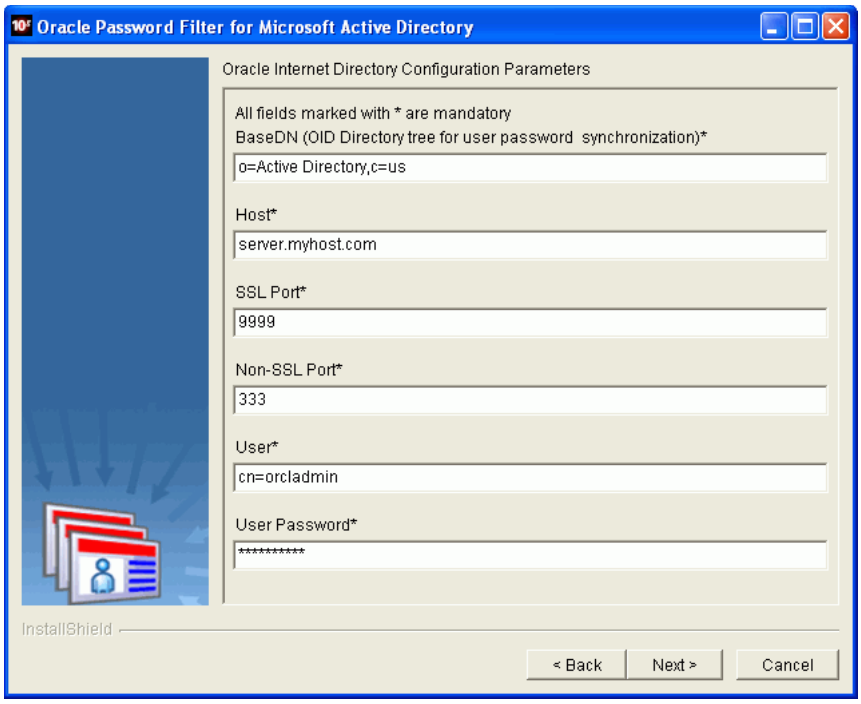
6. On the Microsoft Active Directory Configuration Parameters page, enter values for the following parameters:
   - **Domain**
   - **Base DN**
   - **Port**
   - **Host**
7. Click **Next**. The Microsoft Active Directory Domain Controller Information page displays.



This image shows the Microsoft Active Directory Domain Controller Information page of the Oracle Password Filter for Microsoft Active Directory installation program. This page contains User, User Password, and Log File Path boxes, along with Back, Next, and Cancel buttons.

****

8. On the Microsoft Active Directory Domain Controller Information page, enter values for the following parameters:
   - **User**
   - **User Password**
   - **Log File Path**
9. Click **Next** to continue. The Oracle Internet Directory Configuration Parameters page displays.



This image shows the Oracle Internet Directory Configuration Parameters page of the Oracle Password Filter for Microsoft Active Directory installation program. This page contains Base DN, Host, SSL Port, Non-SSL Port, User, and User Password boxes, along with Back, Next, and Cancel buttons.

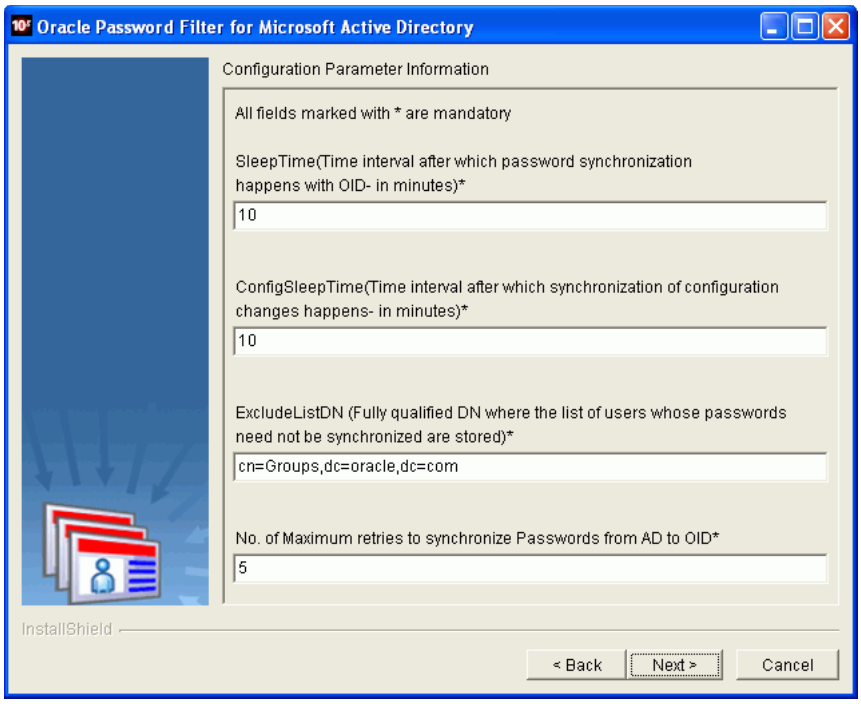
****

10. On the Oracle Internet Directory Configuration Parameters page, enter values for the following parameters:
    - **Base DN**
    - **Host**
    - **SSL Port**
    - **Non-SSL Port**
    - **User**
    - **User Password**

    Important: If you have configured both import and export synchronization between Oracle Internet Directory and Microsoft Active Directory, be sure to enter for the User and User Password parameters the same bind DN and password that are specified in the synchronization profile that imports values from Microsoft Active Directory into Oracle Internet Directory. This is necessary to prevent password updates from looping between Oracle Internet Directory and Microsoft Active Directory.

    ---

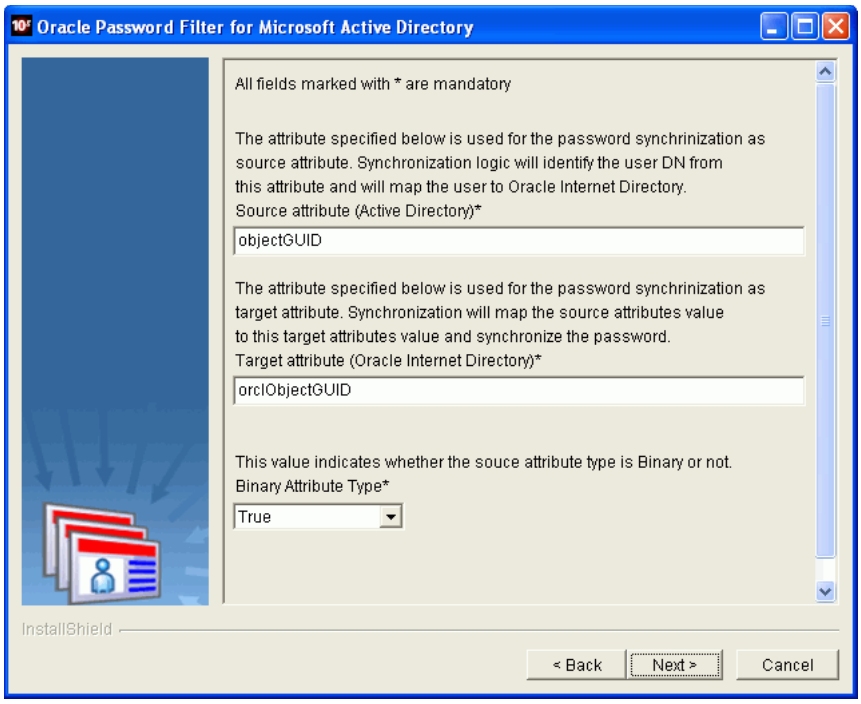
11. Click **Next** to continue. The Oracle Password Filter Configuration Parameters page displays.



This image shows the Oracle Password Filter Configuration Parameters page of the Oracle Password Filter for Microsoft Active Directory installation program. This page contains SleepTime, ConfigSleepTime, ExcludeListDN, and Maximum Retries boxes, along with Back, Next, and Cancel buttons.

****

12. On the Oracle Password Filter Configuration Parameters page, enter values for the following parameters:
    - **SleepTime**: The number of minutes between attempts to synchronize passwords changes between Oracle Internet Directory and Microsoft Active Directory.
    - **ConfigSleepTime**: The number of minutes between attempts to synchronize configuration changes between Oracle Internet Directory and Microsoft Active Directory.
    - **ExcludeListDN**: A fully qualified DN containing a list of users whose passwords should not be synchronized.
    - **Maximum Retries**: Specifies the maximum number of attempts to synchronize a password.
13. Click **Next** to continue. If you chose Advanced on the Installation Options page, the Specify Attributes page displays.



This image shows the Specify Attributes page of the Oracle Password Filter for Microsoft Active Directory installation program. This page contains Source Attribute, Target Attribute, and Binary Attribute Type boxes, along with Back, Next, and Cancel buttons.

****

Perform the following steps for advanced installations:

a. On the Specify Attributes page displays, enter values in the **Source Attribute (Microsoft Active Directory)** and **Target Attribute (Oracle Internet Directory)** boxes for any attributes that you want to synchronize between the two directories. Also, select a value of `true` or `false` from the **Binary Attribute Type** box to specify whether the source attribute type is binary.

b. Click **Next** to continue. The Summary page displays and lists the path where the Oracle Password Filter for Microsoft Active Directory will be installed.

14. On the Summary page, click **Next** to install the Oracle Password Filter.

15. When prompted whether or not to upload schema extensions to Oracle Internet Directory, *always* select **No**. You do not want to upload schema extensions to Oracle Internet Directory because it comes preloaded with the schema extension attributes required for the Microsoft Active Directory Password filter.

    The Reboot Domain Controller page displays.

16. On the Reboot Domain Controller page, click **Next** to restart the computer.

17. Do the following:

    For 32-bit systems

    a. After the computer restarts, log in as an administrator. The remaining configuration tasks for the Oracle Password Filter execute automatically after you log in.

For 64-bit systems

a. After the computer restarts, log in as an administrator.

b. Choose **Start** > **Run**... and type `regedit` in the Run dialog box, then click **OK**.

   The Registry Editor opens.

c. Navigate to the following registry key:

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\
   orclidmpwf\ADConfig
   ```

d. Edit the `ResourceFilePath` as follows, then click **OK**.

   Change
   `C:\\WINDOWS\\system32\\orclmessages.dll`
   to
   `C:\WINDOWS\sysWOW64\orclmessages.dll`

e. Close the Registry Editor.

The Oracle Password Filter for Microsoft Active Directory is now installed.

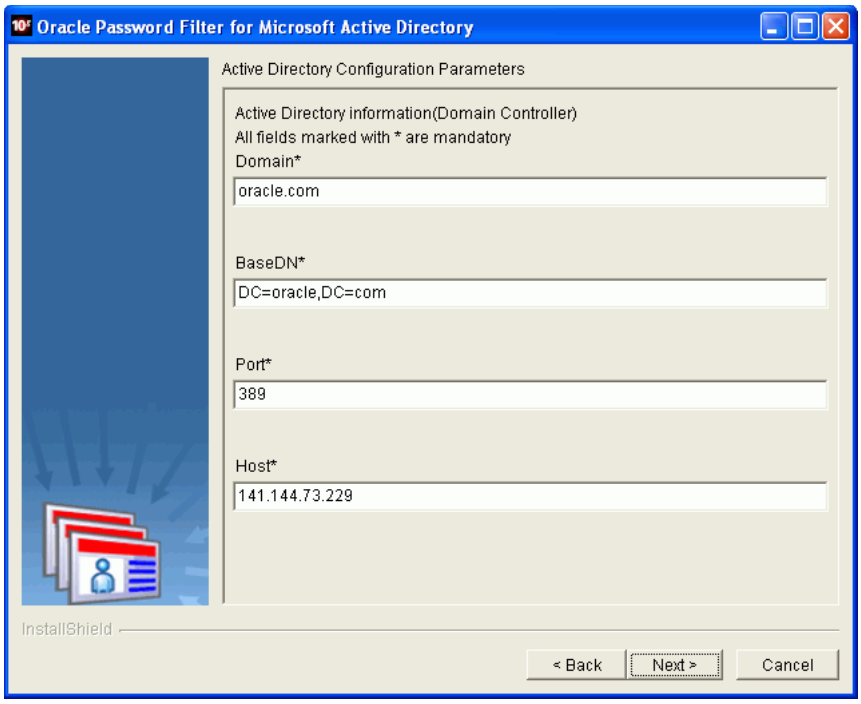
19.5.2 Reconfiguring the Oracle Password Filter for Microsoft Active Directory

In most cases, you should not need to reconfigure the Oracle Password Filter following the installation process. However, you can reconfigure the Oracle Password Filter for Microsoft Active Directory by running the Oracle Password Filter for Microsoft Active Directory installation program.

> **Note:** The Microsoft Active Directory and Oracle Internet Directory configuration parameters listed in the following procedure are described in Table 19–1 and Table 19–2.

To reconfigure the Oracle Password Filter for Microsoft Active Directory:

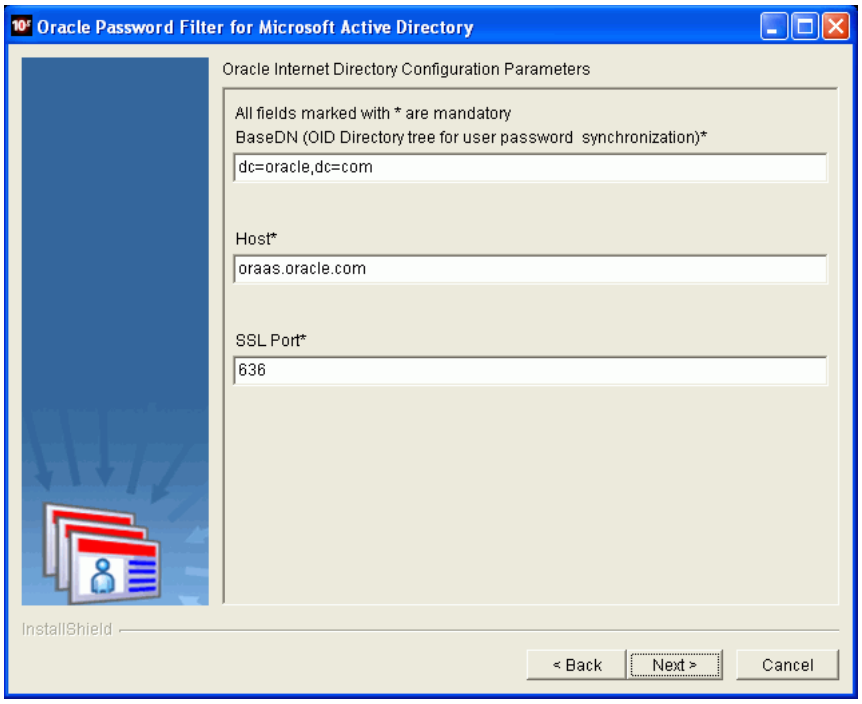
1. Navigate to the directory where you extracted the installation files and double-click **setup.exe**. The Welcome page of the Oracle Password Filter for Microsoft Active Directory configuration program displays, informing you that the installation program will reconfigure the Oracle Password Filter for Microsoft Active Directory.
2. On the Welcome page, click **Next**. The Microsoft Active Directory Configuration Parameters page displays.



This image shows the Microsoft Active Directory Configuration Parameters page of the Oracle Password Filter for Microsoft Active Directory installation program. This page contains Domain, Base DN, Port, and Host boxes, along with Back, Next, and Cancel buttons.

****

3. On the Microsoft Active Directory Configuration Parameters page, modify the following parameters:
   - **Domain**
   - **Base DN**
   - **Port**
   - **Host**
4. Click **Next**. The Oracle Internet Directory Configuration Parameters page displays.



This image shows the Oracle Internet Directory Configuration Parameters page of the Oracle Password Filter for Microsoft Active Directory installation program. This page contains Base DN, Host, and SSL Port boxes, along with Back, Next, and Cancel buttons.

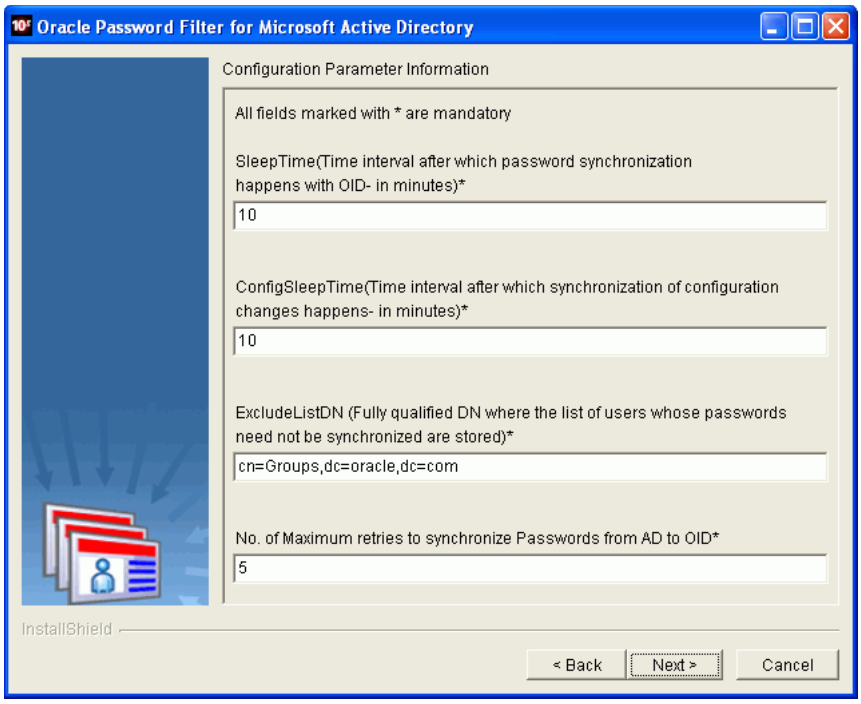
****

5. On the Oracle Internet Directory Configuration Parameters page, modify the following parameters:
   - **Base DN**
   - **Host**
   - **SSL Port**

   Note: At the point of reconfiguring, two configuration set entries exist in Oracle Internet Directory and two instances of the Oracle Internet Directory server are running, each instance with one configuration set entry. Enter the SSL port of the second configuration set entry in the **SSL Port** field.

   ---

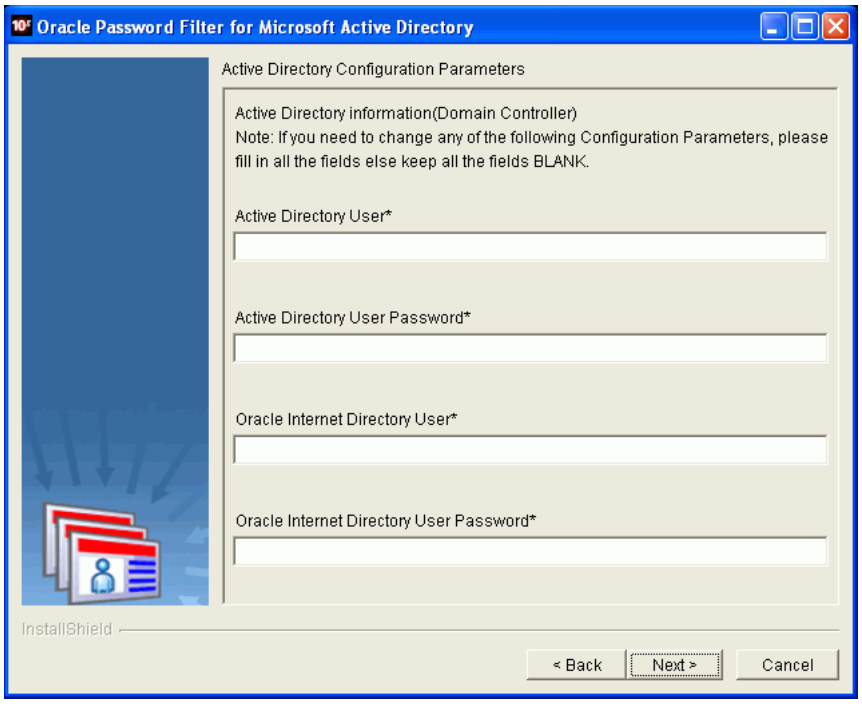
6. Click **Next** to continue. The Oracle Password Filter Configuration Parameters page displays.



This image shows the Oracle Password Filter Configuration Parameters page of the Oracle Password Filter for Microsoft Active Directory installation program. This page contains SleepTime, ConfigSleepTime, ExcludeListDN, and Maximum Retries boxes, along with Back, Next, and Cancel buttons.

****

7. On the Oracle Password Filter Configuration Parameters page, modify the following parameters
   - **SleepTime**: The number of minutes between attempts to synchronize passwords changes between Oracle Internet Directory and Microsoft Active Directory.
   - **ConfigSleepTime**: The number of minutes between attempts to synchronize configuration changes between Oracle Internet Directory and Microsoft Active Directory.
   - **ExcludeListDN**: A fully qualified DN containing a list of users whose passwords should not be synchronized.
   - **Maximum Retries**: Specifies the maximum number of attempts to synchronize a password.
8. Click **Next** to continue. The Oracle Password Filter Users page displays.



This image shows the Oracle Password Filter Configuration Parameters page of the Oracle Password Filter for Microsoft Active Directory installation program. This page contains Active Directory User, Active Directory Password, Oracle Internet Directory User, and Oracle Internet Directory Password boxes, along with Back, Next, and Cancel buttons.

****

9. On the Oracle Password Filter Users page, modify the following parameters:
   - **Microsoft Active Directory User**
   - **Microsoft Active Directory User Password**
   - **Oracle Internet Directory User**
   - **Oracle Internet Directory User Password**

   Important: If you have configured both import and export synchronization between Oracle Internet Directory and Microsoft Active Directory, be sure to enter for the User and User Password parameters the same bind DN and password that are specified in the synchronization profile that imports values from Microsoft Active Directory into Oracle Internet Directory. This is necessary to prevent password updates from looping between Oracle Internet Directory and Microsoft Active Directory.

   ---

10. Click **Next** to continue. The Reconfiguration Completed Successfully page displays.
11. On the Reconfiguration Completed Successfully page, click **Finish** to reconfigure the Oracle Password Filter.

19.6 Removing the Oracle Password Filter for Microsoft Active Directory

This section describes how to remove (uninstall) the Oracle Password Filter for Microsoft Active Directory.

To remove the Oracle Password Filter for Microsoft Active Directory:

1. Open in a text editor the **prepAD.ldif** file, which is located in the directory where you installed the Oracle Password Filter for Microsoft Active Directory. Delete the entries and container listed in the prepAD.ldif file from your Microsoft Active Directory installation.
2. Click the Windows **Start** menu and select **Run**.

   The Run dialog box opens.
3. Enter **regedt32** in the Run dialog box and click **OK**.

   The Registry Editor opens.
4. Navigate to the following registry key:

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\orclidmpwf\OIDConfig
   ```
5. Note the container assigned to the `OidSinkNode` entry. The default value assigned to this entry is `cn=Products,cn=OracleContext`.
6. Select **Control Panel** from the Windows **Start** menu. The Control Panel window displays. In the Control Panel window, select **Add or Remove Programs**. The Add or Remove Programs window displays.
7. In the Add or Remove Programs window, select **Oracle Password Filter for Microsoft Active Directory** from the list of currently installed programs, and then click **Change/Remove**. The Welcome page of the Oracle Password Filter for Microsoft Active Directory installation program displays, informing you that the program will remove the Oracle Password Filter for Microsoft Active Directory.
8. On the Welcome page, click **Next**. The Summary page displays and lists the path from where the Oracle Password Filter for Microsoft Active Directory will be removed.
9. On the Summary page, click **Next**. The Restart Required page appears notifying you that removing the Oracle Password Filter for Microsoft Active Directory requires a restart at the end of the deinstallation process.
10. On the Restart Required page, click **Next**. A final page appears informing you that you must restart your computer. Click **Next** to restart your computer.
11. On the system where Oracle Internet Directory is installed, use Oracle Directory Services Manager or ldapdelete to delete the following entry and its subentries in the **`cn=PWSync,OidSinkNode`** container:

    ```
    CN=Active_Directory_Host, cn=PWSync,OidSinkNode
    ```
12. Create a new text file named deleteOIDSchema.ldif that contains the following entries:

    ```
    dn: cn=subschemasubentry
    changetype: modify
    delete: objectclasses
    objectclasses: ( 2.16.840.1.113894.8.2.1002 NAME 'adconfig' SUP top STRUCTURAL
    MUST ( cn ) MAY ( ADBaseDN $ deleteomain $ ADHost $ ADPort $ Log $
    ResourceFilePath ) )
    ```

```
dn: cn=subschemasubentry
changetype: modify
delete: objectclasses
objectclasses: ( 2.16.840.1.113894.8.2.1001 NAME 'oidconfig' SUP top STRUCTURAL
MUST ( cn ) MAY ( OIDBaseDN $ OIDHost $ OIDPort $ passwdattr $ MSDEDSN $
OIDObjectClass $ OIDLog $ ExcludeListDN $ MAX_RETRIES $ OIDSSLType $
OIDWalletLoc $ OidSinkNode $ SleepTime $ stop $ ConfigSleepTime $
OIDConfigSynchKey ) )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1001  NAME 'OIDBaseDN' DESC 'OID Base
Search DN' SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1002  NAME 'OIDHost' DESC 'OID Host'
SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1003  NAME 'OIDPort' DESC 'OID Port'
SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1004  NAME 'passwdattr' DESC 'Pass
Attribute' SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1005  NAME 'MSDEDSN' DESC 'DB DSN'
SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1006  NAME 'OIDObjectClass' DESC 'AD
Object Class' SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1007  NAME 'OIDLog' DESC 'OID Log'
SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1008  NAME 'ExcludeListDN' DESC
'Exclude List' SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
```

```
attributetypes: ( 2.16.840.1.113894.8.1.1009  NAME 'MAX_RETRIES' DESC 'Max
Retries' SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1010  NAME 'OIDSSLType' DESC 'OID SSL
Type' SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1011  NAME 'OIDWalletLoc' DESC 'OID
Wallet Loc' SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1012  NAME 'OidSinkNode' DESC 'Config
Sync Node' SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1013  NAME 'SleepTime' DESC 'Sleep Time
for store thread' SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1014 NAME 'stop' DESC 'Stop flag for
store thread' SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1015 NAME 'ConfigSleepTime' DESC 'Sleep
Time for config thread' SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 22.16.840.1.113894.8.1.1016 NAME 'OIDConfigSynchKey' DESC
'Config Sync key' SYNTAX '1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1017 NAME 'ADBaseDN' SYNTAX
'1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1018 NAME 'ADPort' SYNTAX
'1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
```

```
attributetypes: ( 2.16.840.1.113894.8.1.1019 NAME 'ADHost' SYNTAX
'1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1020 NAME 'ADDomain' SYNTAX
'1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1021 NAME 'Log' SYNTAX
'1.3.6.1.4.1.1466.115.121.1.15' )

dn: cn=subschemasubentry
changetype: modify
delete: attributetypes
attributetypes: ( 2.16.840.1.113894.8.1.1022 NAME 'ResourceFilePath' SYNTAX
'1.3.6.1.4.1.1466.115.121.1.15' )
```

13. Use an `ldapmodify` command to load the deleteOIDSchema.ldif file:

```
$ORACLE_HOME/bin/ldapmodify -h OID host -p OID port \
-D binddn -q -f deleteOIDSchema.ldif
```

Note: You will be prompted for the password.

20

Integrating with Oracle Directory Server Enterprise Edition (Sun Java System Directory Server)

This chapter outlines the procedures for integrating Oracle Identity Management with Oracle Directory Server Enterprise Edition (previously known as Sun Java System Directory Server, and, before that, SunONE iPlanet). It contains these topics:

- Verifying Synchronization Requirements for Oracle Directory Server Enterprise Edition
- Configuring Basic Synchronization with Oracle Directory Server Enterprise Edition
- Configuring Advanced Integration with Oracle Directory Server Enterprise Edition

---

Note: Before continuing with this chapter, you should be familiar with the concepts presented in previous chapters. The following chapters in particular are important:

- Chapter 1, "Introduction to Oracle Identity Management Integration"
- Chapter 4, "Managing the Oracle Directory Integration Platform"
- Chapter 5, "Understanding the Oracle Directory Synchronization Service"
- Chapter 16, "Connected Directory Integration Concepts and Considerations"

If you are configuring a demonstration of integration with Oracle Directory Server Enterprise Edition / Sun Java System Directory Server, then see the Oracle By Example series for Oracle Identity Management Release 11*g* Release 1 (11.1.1), available on Oracle Technology Network at `http://www.oracle.com/technology/`

20.1 Verifying Synchronization Requirements for Oracle Directory Server Enterprise Edition

Before configuring basic or advanced synchronization with Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server), ensure that your environment meets the necessary synchronization requirements by following the

instructions in "Verifying Synchronization Requirements" on page 17-1. Before synchronizing with Oracle Directory Server Enterprise Edition, you must also perform the following steps:

- When creating a user account in Oracle Directory Server Enterprise Edition with sufficient privileges to perform import and export operations, be sure to assign sufficient permissions to read the tombstone.
- Enable change logging on Oracle Directory Server Enterprise Edition.
- Enable the Retro Change Log plug-in.

20.2 Configuring Basic Synchronization with Oracle Directory Server Enterprise Edition

You use the `expressSyncSetup` command to quickly establish synchronization between the Oracle back-end directory and Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server). The `expressSyncSetup` command uses default settings to automatically perform all required configurations, and also creates two synchronization profiles, one for import and one for export. To use the `expressSyncSetup` command to synchronize with Oracle Directory Server Enterprise Edition, refer to "Creating Import and Export Synchronization Profiles Using expressSyncSetup" on page 17-2.

20.3 Configuring Advanced Integration with Oracle Directory Server Enterprise Edition

When you install Oracle Directory Integration Platform, sample import and export synchronization profiles are automatically created for each of the supported directories that Oracle Directory Integration Platform can connect to. The sample synchronization profiles created for Oracle Directory Server Enterprise Edition are:

- `iPlanetImport`—The profile for importing changes from Oracle Directory Server Enterprise Edition to the Oracle back-end directory
- `iPlanetExport`—The profile for exporting changes from the Oracle back-end directory to Oracle Directory Server Enterprise Edition

You can also use the `expressSyncSetup` command or Oracle Enterprise Manager Fusion Middleware Control to create additional synchronization profiles. The import and export synchronization profiles created during the install process or with the `expressSyncSetup` command are only intended as a starting point for you to use when deploying your integration of the Oracle back-end directory and Oracle Directory Server Enterprise Edition. Because the default synchronization profiles are created using predefined assumptions, you must further customize them for your environment by performing the following steps in the order listed:

- Step 1: Plan Your Integration
- Step 2: Configure the Realm
- Step 3: Customize the ACLs
- Step 4: Customize Attribute Mappings
- Step 5: Customize the Oracle Directory Server Enterprise Edition (Sun Java System Directory Server) Connector to Synchronize Deletions
- Step 6: Synchronize Passwords

- Step 7: Synchronizing in SSL Mode
- Step 8: Configure the Oracle Directory Server Enterprise Edition (Sun Java System Directory Server) External Authentication Plug-in
- Step 9: Perform Post-Configuration and Administrative Tasks

20.3.1 Step 1: Plan Your Integration

Plan your integration by reading Chapter 16, "Connected Directory Integration Concepts and Considerations", particularly "Oracle Directory Server Enterprise Edition (Sun Java System Directory Server) Integration Concepts" on page 16-30. Be sure to create a new profile by copying the existing Oracle Directory Server Enterprise Edition or Sun Java System Directory Server template profile by following the instructions in "Creating Synchronization Profiles" on page 7-1.

20.3.2 Step 2: Configure the Realm

Configure the realm by following the instructions in "Configuring the Realm" on page 17-8.

20.3.3 Step 3: Customize the ACLs

Customize ACLs as described in "Customizing Access Control Lists" on page 17-9.

20.3.4 Step 4: Customize Attribute Mappings

When integrating with Oracle Directory Server Enterprise Edition, the following attribute-level mapping is mandatory for all objects:

```
Targetdn:1: :person:orclsourceobjectdn: : orclSUNOneobject:
```

Example 20–1 Attribute-Level Mapping for the User Object in Oracle Directory Server Enterprise Edition (Sun Java System Directory Server)

```
Cn:1: :person: cn: :person:
sn:1: :person: sn: :person:
```

Example 20–2 Attribute-Level Mapping for the Group Object in Oracle Directory Server Enterprise Edition (Sun Java System Directory Server)

```
Cn:1: :groupofname: cn:groupofuniquenames
```

In the preceding examples, `Cn` and `sn` from Oracle Directory Server Enterprise Edition are mapped to `cn` and `sn` in the Oracle back-end directory.

Customize the attribute mappings by following the instructions in "Customizing Mapping Rules" on page 17-12.

20.3.5 Step 5: Customize the Oracle Directory Server Enterprise Edition (Sun Java System Directory Server) Connector to Synchronize Deletions

If you want to synchronize deletions, and the mapping rules have mandatory attributes, then be sure that the tombstone is configured correctly.

To verify that the tombstone is configured in Oracle Directory Server Enterprise Edition, execute the following command:

```
$ORACLE_HOME/bin/ldapsearch -h connected_directory_host \
-p connected_directory_port -D  connected_directory_account -q \
```

```
-b source_domain -s sub "objectclass=nstombstone"
```

Note: You will be prompted for the password.

This returns information on all deleted entries.

See Also: Oracle Directory Server Enterprise Edition or Sun Java System Directory Server documentation for details about configuring tombstones

Note: Tombstones are automatically configured for Oracle Directory Server Enterprise Edition if replication is enabled.

20.3.6 Step 6: Synchronize Passwords

The Oracle back-end directory and Oracle Directory Server Enterprise Edition support the same set of password hashing techniques. To synchronize passwords between Oracle Internet Directory and Oracle Directory Server Enterprise Edition, ensure that SSL server authentication mode is configured for both directories and that the following mapping rule exists in the mapping file:

```
Userpassword: : :person:userpassword: :person
```

If your Oracle back-end directory is Oracle Unified Directory, Oracle Directory Integration Platform does not support password synchronization to Oracle Directory Server Enterprise Edition from Oracle Unified Directory. One-way password synchronization from Oracle Directory Server Enterprise Edition to an Oracle Unified Directory back-end directory is supported.

20.3.7 Step 7: Synchronizing in SSL Mode

Configure Oracle Directory Server Enterprise Edition for synchronization in SSL mode by following the instructions in "Configuring the Connected Directory Connector for Synchronization in SSL Mode" on page 17-13.

20.3.8 Step 8: Configure the Oracle Directory Server Enterprise Edition (Sun Java System Directory Server) External Authentication Plug-in

Configure the Oracle Directory Server Enterprise Edition (Sun Java System Directory Server) external authentication plug-in by following the instructions in on page 17-16 "Configuring External Authentication Plug-ins".

20.3.9 Step 9: Perform Post-Configuration and Administrative Tasks

Read Chapter 23, "Managing Integration with a Connected Directory" for information on post-configuration and ongoing administration tasks.

21

Integrating with IBM Tivoli Directory Server

This chapter outlines the procedures for integrating Oracle Identity Management with IBM Tivoli Directory Server. It contains these topics:

- Verifying Synchronization Requirements for IBM Tivoli Directory Server
- Configuring Basic Synchronization with IBM Tivoli Directory Server
- Configuring Advanced Integration with IBM Tivoli Directory Server

---

Note: Before continuing with this chapter, you should be familiar with the concepts presented in previous chapters. The following chapters in particular are important:

- Chapter 1, "Introduction to Oracle Identity Management Integration"
- Chapter 4, "Managing the Oracle Directory Integration Platform"
- Chapter 5, "Understanding the Oracle Directory Synchronization Service"
- Chapter 16, "Connected Directory Integration Concepts and Considerations"

If you are configuring a demonstration of integration with IBM Tivoli Directory Server, then see the Oracle By Example series for Oracle Identity Management Release 11*g* Release 1 (11.1.1), available on Oracle Technology Network at
`http://www.oracle.com/technology/`

21.1 Verifying Synchronization Requirements for IBM Tivoli Directory Server

Before configuring basic or advanced synchronization with IBM Tivoli Directory Server, ensure that your environment meets the necessary synchronization requirements by following the instructions in "Verifying Synchronization Requirements" on page 17-1. Before synchronizing with IBM Tivoli Directory Server, you must also perform the following steps:

- When creating a user account in IBM Tivoli Directory Server with sufficient privileges to perform import and export operations, be sure to assign sufficient permissions to read the tombstone
- Enable change logging on IBM Tivoli Directory Server

21.2 Configuring Basic Synchronization with IBM Tivoli Directory Server

You use the `expressSyncSetup` command to quickly establish synchronization between the Oracle back-end directory and IBM Tivoli Directory Server. The `expressSyncSetup` command uses default settings to automatically perform all required configurations, and also creates two synchronization profiles, one for import and one for export. To use the `expressSyncSetup` command to synchronize with IBM Tivoli Directory Server, refer to "Creating Import and Export Synchronization Profiles Using expressSyncSetup" on page 17-2.

21.3 Configuring Advanced Integration with IBM Tivoli Directory Server

When you install Oracle Directory Integration Platform, sample import and export synchronization profiles are automatically created for each of the supported third-party directories. The sample synchronization profiles created for IBM Tivoli Directory Server are:

- `TivoliImport`—The profile for importing changes from IBM Tivoli Directory Server to the Oracle back-end directory.
- `TivoliExport`—The profile for exporting changes from the Oracle back-end directory to IBM Tivoli Directory Server

You can also use the `expressSyncSetup` command to create additional synchronization profiles. The import and export synchronization profiles that you either created during the install process or with `expressSyncSetup` are only intended as a starting point for you to use when deploying your integration of the Oracle back-end directory and an IBM Tivoli Directory Server. Because the default synchronization profiles are created using predefined assumptions, you must further customize them for your environment by performing the following steps in the order listed:

- Step 1: Plan Your Integration
- Step 2: Configure the Realm
- Step 3: Customize the ACLs
- Step 4: Customize Attribute Mappings
- Step 5: Customize the IBM Tivoli Directory Server Connector to Synchronize Deletions
- Step 6: Synchronize Passwords
- Step 7: Synchronize in SSL Mode
- Step 8: Configure the IBM Tivoli Directory Server External Authentication Plug-in
- Step 9: Perform Post-Configuration and Administrative Tasks

21.3.1 Step 1: Plan Your Integration

Plan your integration by reading Chapter 16, "Connected Directory Integration Concepts and Considerations", particularly "IBM Tivoli Directory Server Integration Concepts" on page 16-30. Be sure to create a new profile by copying the existing IBM Tivoli Directory Server template profile by following the instructions in "Creating Synchronization Profiles" on page 7-1.

21.3.2 Step 2: Configure the Realm

Configure the realm by following the instructions in "Configuring the Realm" on page 17-8.

21.3.3 Step 3: Customize the ACLs

Customize ACLs as described in "Customizing Access Control Lists" on page 17-9.

21.3.4 Step 4: Customize Attribute Mappings

When integrating with IBM Tivoli Directory Server, the following attribute-level mapping is mandatory for all objects:

```
targetdn: : :top:orclSourceObjectDN: :orclTDSObject:
```

Example 21–1 Attribute-Level Mapping for the User Object in IBM Tivoli Directory Server

```
Cn:1: :person: cn: :person:
sn: : :person: sn: :person:
```

Example 21–2 Attribute-Level Mapping for the Group Object in IBM Tivoli Directory Server

```
Cn:1: :groupofname: cn:groupofuniquenames
```

In the preceding examples, `Cn` and `sn` from IBM Tivoli Directory Server are mapped to `cn` and `sn` in the Oracle back-end directory.

If you specify anything other than the RDN attribute as a required attribute in the mapping file, those changes will not be synchronized. This is due to a limitation in IBM Tivoli Directory Server where changes do not appear as deletions in the changelog when tombstones are enabled.

Customize the attribute mappings by following the instructions in "Customizing Mapping Rules" on page 17-12.

21.3.5 Step 5: Customize the IBM Tivoli Directory Server Connector to Synchronize Deletions

If you want to synchronize deletions, you must ensure tombstones are not enabled in IBM Tivoli Directory Server. To check if tombstones are enabled, execute the following command:

```
ldapsearch -h connected_directory_host -p connected_directory_port \
-D binddn -q \
-b "cn=Directory, cn=RDBM Backends, cn=IBM
Directory, cn=Schemas, cn=Configuration" -s base "objectclass=*"
ibm-slapdTombstoneEnabled
```

Note: You will be prompted for the password.

This command returns information on all deleted entries.

See Also: IBM Tivoli Directory Server documentation for details about configuring tombstones.

21.3.6 Step 6: Synchronize Passwords

The Oracle back-end directory and IBM Tivoli Directory Server support the same set of password hashing techniques. To synchronize passwords from IBM Tivoli Directory Server to the Oracle back-end directory, ensure that SSL server authentication mode is configured for both directories and that the following mapping rule exists in the mapping file:

```
Userpassword: : :person:userpassword: :person
```

Two-way password synchronization is not supported from an Oracle Unified Directory back-end directory or an Oracle Directory Server Enterprise Edition back-end directory. Two-way password synchronization is only supported if the back-end directory is Oracle Internet Directory.

21.3.7 Step 7: Synchronize in SSL Mode

Configure IBM Tivoli Directory Server for synchronization in SSL mode by following the instructions in "Configuring the Connected Directory Connector for Synchronization in SSL Mode" on page 17-13.

21.3.8 Step 8: Configure the IBM Tivoli Directory Server External Authentication Plug-in

Perform the following steps to configure an IBM Tivoli Directory Server external authentication plug-in:

1. Add the configuration entries for the external authentication plug-in for IBM Tivoli Directory Server to the Oracle back-end directory by performing the following steps:

   Note: The wallet referred to in the configuration entries for the external authentication plug-in for IBM Tivoli Directory Server is ORACLE wallet. Accordingly, use Oracle wallet commands to add and remove certificates from the wallet. JKS commands are used only for the certificates that Oracle Directory Integration Platform uses.

   a. Copy the following entries to an LDIF file, for example, *input.ldif*:

```
dn: cn=oidexplg_compare_tivoli,cn=plugin,cn=subconfigsubentry
cn: oidexplg_compare_tivoli
objectclass: orclPluginConfig
objectclass: top
orclpluginname: oidexplg
orclplugintype: operational
orclpluginkind: Java
orclplugintiming: when
orclpluginldapoperation: ldapcompare
orclpluginsecuredflexfield;walletpwd: password
orclpluginsecuredflexfield;walletpwd2: password
orclpluginversion: 1.0.1
orclpluginisreplace: 1
orclpluginattributelist: userpassword
orclpluginentryproperties:
(!(&(objectclass=orclTDSobject)(objectclass=orcluserv2)))
```

```
orclpluginflexfield;host2: host.domain.com
orclpluginflexfield;port2: 636
orclpluginflexfield;isssl2: 1
orclpluginflexfield;host: host.domain.com
orclpluginflexfield;walletloc2: /location/wallet
orclpluginflexfield;port: 389
orclpluginflexfield;walletloc: /tmp
orclpluginflexfield;isssl: 0
orclpluginflexfield;isfailover: 0
orclpluginclassreloadenabled: 0
orclpluginenable: 0
orclpluginsubscriberdnlist: cn=users,dc=us,dc=oracle,dc=com

dn: cn=oidexplg_bind_tivoli,cn=plugin,cn=subconfigsubentry
cn: oidexplg_bind_tivoli
objectclass: orclPluginConfigobjectclass: top
orclpluginname: oidexplg
orclplugintype: operational
orclpluginkind: Java
orclplugintiming: when
orclpluginldapoperation: ldapbind
orclpluginversion: 1.0.1
orclpluginisreplace: 1
orclpluginentryproperties:
(!(&(objectclass=orclTDSobject)(objectclass=orcluserv2)))
orclpluginclassreloadenabled: 0
orclpluginflexfield;walletloc2: /location/wallet
orclpluginflexfield;port: 389
orclpluginflexfield;walletloc: /tmp
orclpluginflexfield;isssl: 0
orclpluginflexfield;isfailover: 0
orclpluginflexfield;host2: host.domain.com
orclpluginflexfield;port2: 636
orclpluginflexfield;isssl2: 1
orclpluginflexfield;host: host.domain.com
orclpluginenable: 0
orclpluginsecuredflexfield;walletpwd: password
orclpluginsecuredflexfield;walletpwd2: password
orclpluginsubscriberdnlist:
cn=users,dc=us,dc=oracle,dc=com
```

b. Copy the entries in the LDIF file to the Oracle back-end directory using a command similar to the following:

```
ldapadd -h HOST -p PORT -D binddn -q -v -f input.ldif
```

Note: You will be prompted for the password.

2. Use the instructions in "Configuring External Authentication Plug-ins" on page 17-16 to configure the plug-in.

21.3.9 Step 9: Perform Post-Configuration and Administrative Tasks

Read Chapter 23, "Managing Integration with a Connected Directory" for information on post-configuration and ongoing administration tasks.

22

Integrating with Novell eDirectory or OpenLDAP

This chapter outlines the procedures for integrating Oracle Identity Management with Novell eDirectory or OpenLDAP in a production environment. It contains these topics:

- Verifying Synchronization Requirements for Novell eDirectory or OpenLDAP
- Configuring Basic Synchronization with Novell eDirectory or OpenLDAP
- Configuring Advanced Integration with Novell eDirectory or OpenLDAP

Notes: Before continuing with this chapter, you should be familiar with the concepts presented in previous chapters. The following chapters in particular are important:

- Chapter 1, "Introduction to Oracle Identity Management Integration"
- Chapter 4, "Managing the Oracle Directory Integration Platform"
- Chapter 5, "Understanding the Oracle Directory Synchronization Service"
- Chapter 16, "Connected Directory Integration Concepts and Considerations"

Synchronization is supported between Oracle Fusion Middleware 11g Release 1 (11.1.1) or later and Novell eDirectory 8.6.2 or later or OpenLDAP 2.2.

22.1 Verifying Synchronization Requirements for Novell eDirectory or OpenLDAP

Before configuring basic or advanced synchronization with Novell eDirectory or OpenLDAP, ensure that your environment meets the necessary synchronization requirements by following the instructions in "Verifying Synchronization Requirements" on page 17-1.

Note: To reconcile correctly, additions and deletions must be performed from only one of the synchronized directories. In other words, you can perform additions and deletions from the Oracle back-end directory or eDirectory/OpenLDAP, but not both. However, modifications can be performed from either directory.

22.2 Configuring Basic Synchronization with Novell eDirectory or OpenLDAP

You can use the `expressSyncSetup` command to quickly establish synchronization between the Oracle back-end directory and Novell eDirectory or OpenLDAP. The `expressSyncSetup` command uses default settings to automatically perform all required configurations. To use the `expressSyncSetup` command to synchronize with Novell eDirectory or OpenLDAP, refer to "Creating Import and Export Synchronization Profiles Using expressSyncSetup" on page 17-2.

22.2.1 Synchronizing Multiple Profiles from eDirectory or OpenLDAP to One Oracle Back-end Directory Container

When synchronizing multiple profiles from eDirectory or OpenLDAP to one Oracle back-end directory container, you must filter out only the specific users to be reconciled to prevent the reconciliation process from inadvertently deleting users. You can filter out only the specific users to be reconciled by performing either of the following steps:

- Modify the mapping rule so each profile creates the user in a different container. Refer to "Customizing Mapping Rules" on page 17-12 for more information.
- Modify the reconciliation rules in the mapping file to synchronize only a specific subset of users. Refer to "How Do I Define a Reconciliation Rule?" on page 22-5 for more information.

22.3 Configuring Advanced Integration with Novell eDirectory or OpenLDAP

When you install Oracle Directory Integration Platform, sample import and export synchronization profiles are automatically created for each of the supported directories. The sample synchronization profiles created for Novell eDirectory are:

- `Novell eDirectoryImp`—The profile for importing changes from Novell eDirectory to the Oracle back-end directory.
- `Novell eDirectoryExp`—The profile for exporting changes from the Oracle back-end directory to Novell eDirectory.

The sample synchronization profiles created for OpenLDAP are:

- `OpenLDAPImport`—The profile for importing changes from OpenLDAP to the Oracle back-end directory.
- `OpenLDAPExport`—The profile for exporting changes from the Oracle back-end directory to OpenLDAP.

You can also use the `expressSyncSetup` command or Oracle Enterprise Manager Fusion Middleware Control to create additional synchronization profiles. The import and export synchronization profiles created during the install process or with `expressSyncSetup` are only intended as a starting point for you to use when deploying your integration of the Oracle back-end directory and Novell eDirectory or OpenLDAP. Because the default synchronization profiles are created using predefined assumptions, you must further customize them for your environment by performing the following steps in the order listed:

- Step 1: Plan Your Integration
- Step 2: Configure the Realm

22.3.1 Step 1: Plan Your Integration

Plan your integration by reading Chapter 16, "Connected Directory Integration Concepts and Considerations", particularly "Novell eDirectory and OpenLDAP Integration Concepts" on page 16-31. Be sure to create a new profile by copying the existing eDirectory or OpenLDAP template profile by following the instructions in "Creating Synchronization Profiles" on page 7-1.

22.3.2 Step 2: Configure the Realm

Configure the realm by following the instructions in "Configuring the Realm" on page 17-8.

22.3.3 Step 3: Customize the Search Filter to Retrieve Information from Novell eDirectory or OpenLDAP

By default, the Novell eDirectory or OpenLDAP Connector retrieves changes to all objects in the container based on the `modifytimestamp` attribute. If you are interested in retrieving changes to specific types of objects, such as changes to users and groups, then you should configure an LDAP search filter. This filter screens out changes that are not required when the Novell eDirectory or OpenLDAP Connector queries Novell eDirectory or OpenLDAP. The filter is stored in the connected directory matching filter attribute (`orclodipcondirmatchingfilter`) in the synchronization profile.

The Novell eDirectory and OpenLDAP sample import profiles are configured to retrieve changes to users, groups, and container objects from Novell eDirectory and OpenLDAP, respectively. Computers are not retrieved. The value of the `searchfilter` attribute is set as follows:

```
searchfilter=(&(!(modifiersname=connected_dir_account))
(|(objectclass=domain)(objectclass=organizationalunit)
(objectclass=organization)(objectclass=person) (objectclass=groupofnames)))
```

You use the `update` operation of the manageSyncProfiles command to update the `searchfilter` attribute if you want to synchronize entries other than users or

groups. For example, the following command updates the `searchfilter` attribute to synchronize only users and groups:

```
manageSyncProfiles -operation update -profile profile_name
odip.profile.condirfilter searchfilter=
(|(objectclass=groupofnames)(objectclass=person))
```

Notes:

- All attributes specified in the `searchfilter` attribute should be configured as indexed attributes in Novell eDirectory or OpenLDAP.
- Refer to "Managing Synchronization Profiles Using manageSyncProfiles" on page 7-16 for more information about the `manageSyncProfiles` command.

See Also: The appendix on the LDAP filter definition in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for instructions on configuring an LDAP search filter.

22.3.4 Step 4: Customize the ACLs

Customize ACLs as described in "Customizing Access Control Lists" on page 17-9.

22.3.5 Step 5: Customize Attribute Mappings

When integrating with Novell eDirectory, the following attribute-level mapping is mandatory for all objects:

```
GUID:1: : :orclNDSObjectGUID: :orclndsObject:bin2b64(guid)
Modifytimestamp:1 : : :orclsourcemodifytimestamp: :orclndsobject:
Createtimestamp:1 : : :orclsourcecreatetimestamp: :orclndsobject:
Targetdn:1: : :orclsourceobjectdn: : orclndsobject:
```

When integrating with OpenLDAP, the following attribute-level mapping is mandatory for all objects:

```
entryuuid:1: : : orclOpenLdapEntryUUID: : orclOpenLdapObject
Modifytimestamp:1 : : :orclsourcemodifytimestamp: : orclOpenLdapObject
Createtimestamp:1 : : :orclsourcecreatetimestamp: : orclOpenLdapObject
Targetdn:1: : :orclsourceobjectdn: : orclOpenLdapObject:
```

Example 22–1 Attribute-Level Mapping for the User Object in Novell eDirectory or OpenLDAP

```
Cn:1: : :person: cn: :person:
sn:1: : :person: sn: :person:
```

Example 22–2 Attribute-Level Mapping for the Group Object in Novell eDirectory or OpenLDAP

```
Cn:1: : :groupofname: cn:groupofuniquenames
```

In the preceding examples, `Cn` and `sn` from Novell eDirectory or OpenLDAP are mapped to `cn` and `sn` in the Oracle back-end directory.

Customize the attribute mappings by following the instructions in "Customizing Mapping Rules" on page 17-12.

22.3.6 Step 6: Customize the Novell eDirectory or OpenLDAP Connector to Synchronize Deletions

Synchronizing deletions from Novell eDirectory or OpenLDAP in the Oracle back-end directory is handled with the reconciliation approach, as described in "Synchronizing from Novell eDirectory or OpenLDAP to the Oracle Back-end Directory" on page 16-32. Because the reconciliation process is time and CPU intensive, by default, reconciliation occurs at a 3600 second (or 1 hour) interval. You can modify the length of this interval according to your environment by using the `manageSycnProfiles` command and `-params` option to modify the `odip.profile.reconciliationtimeinterval` parameter.

To avoid decreased performance on the server when synchronizing deletions from Novell eDirectory or OpenLDAP in the Oracle back-end directory, you can customize the comparison to search specific subsets of the DIT. You specify the subset search criteria as part of the map file by using the `ReconciliationRules` keyword.

The default reconciliation rules for Novell eDirectory are as follows:

```
inetorgperson:cn:*
groupofnames:cn:*
```

The default reconciliation rules for OpenLDAP are as follows:

```
inetorgperson:cn:*
groupofuniquenames:cn:*
```

The preceding rules specify that the search criteria be applied in the following two steps:

1. Search for all entries in the `inetorgperson` object class. You can also specify different subsets within this rule according to the attribute values.
2. Search for all entries in the `groupofnames` object class in Novell eDirectory or in the `groupofuniquenames` object class in OpenLDAP.

22.3.6.1 How Do I Define a Reconciliation Rule?

You define a reconciliation rule with one object class, one attribute, and any number of values. You can use any attribute that is synchronized with the Oracle back-end directory to define a reconciliation rule. However, you must observe the following two requirements:

- The attribute of the specified object class must be defined in the mapping rules.
- The corresponding Oracle back-end directory attribute must be indexed.

For example, consider the following reconciliation rule:

```
myobjclass:myattr:val1:val2:val3
```

In the preceding reconciliation rule, the name of the object class is `myobjclass` and the name of the attribute is `myattr`. You can assign values of `val1`, `val2`, or `val3` to the `myattr` attribute. To use the `myattr` attribute, the following mapping rule must be defined:

```
myattr: : : myobjclass:attr: :objclass:
```

The preceding mapping rule defines the `myattr` attribute in the `myobjclass` object class, and `attr` is the corresponding Oracle back-end directory attribute that should be indexed.

22.3.6.2 How are Reconciliation Rules Used to Synchronize Deletions?

Defining reconciliation rules generates search filters that query Novell eDirectory or OpenLDAP to determine the number of deleted entries. For example, with the `myobjclass` and `attr` reconciliation rule example in the previous section, the following search filters are generated in Novell eDirectory or OpenLDAP:

- `(&(objectclass= myobjclass) (createtimestamp<=orclodipreconciliationtimestamp) (myattr=val1))`
- `(&(objectclass= myobjclass) (createtimestamp<= orclodipreconciliationtimestamp) (myattr=val2))`
- `(&(objectclass= myobjclass)(createtimestamp<= orclodipreconciliationtimestamp)(myattr=val3))`

The reconciliation rule and mapping rule also generate corresponding filters in the Oracle back-end directory. For example, the following Oracle back-end directory filters are generated for the `myobjclass` and `attr` reconciliation rule:

- `(&(objectclass= objclass) (orclndsobjectguid=*)(orclSourceCreateTimeStamp<= orclodipreconciliationtimestamp)(attr=val1))`
- `(&(objectclass= objclass) (orclndsobjectguid=*)(orclSourceCreateTimeStamp<= orclodipreconciliationtimestamp)(attr=val2))`
- `(&(objectclass= objclass) (orclndsobjectguid=*)(orclSourceCreateTimeStamp<= orclodipreconciliationtimestamp)(attr=val3))`

22.3.7 Step 7: Specify Synchronization Parameters for the Advanced Configuration Information Attribute

The Advanced Configuration Information (`orclodipAgentConfigInfo`) attribute in a synchronization profile stores any additional configuration information needed by a connector to synchronize the Oracle back-end directory with a connected directory. You can use the `SearchDeltaSize` and `SkipErrorToSyncNextChange` parameters with any connected directory.

For Novell eDirectory and OpenLDAP, you can also use the parameters listed in Table 22–1 to specify additional configuration information.

> **Tip:** Refer to the "Advanced" section on page 7-5 for a description of all Advanced Configuration parameters for synchronization profiles.

Table 22–1 Novell eDirectory and OpenLDAP Synchronization Parameters for the Advanced Configuration Information Attribute

Parameter	Description
`AttributeType`	Indicates the type of the `UniqueAttribute` parameter. You assign to this parameter a value of `Binary` for Novell eDirectory or `nonBinary` for OpenLDAP. This parameter is used to obtain the corresponding Oracle back-end directory attribute for the attribute that is defined in the mapping file.
`SearchTimeDeltaInSeconds`	This parameter is applicable only for eDirectory and OpenLDAP, which handle synchronization based on timestamps and do not support changelog. Search Time Delta Size in seconds determines the time interval for processing changes during each synchronization cycle iteration. The default value is 3600. The number of iterations performed during each synchronization cycle depend on the number of pending changes. For example, if the Search Time Delta In Seconds parameter is set to 60 and there are changes pending for about one minute, synchronization will require a single iteration. If changes are pending for three minutes, synchronization will require three iterations. When the number of changes per minute is small, you will experience better synchronization efficiency by setting Search Time Delta Size in seconds to a higher value. Be sure the value you set for the Search Time Delta In Seconds parameter does not exceed the LDAP search limit of the connected directory server. Otherwise, you may receive an error during synchronization and some changes may not be processed.
`CheckAllEntries`	Determines how deleted entries in Novell eDirectory or OpenLDAP are synchronized with the Oracle back-end directory. If you assign a value of `true` to this parameter, the Oracle Directory Integration Platform identifies deleted entries by performing a linear comparison between the entries in the Oracle back-end directory and Novell eDirectory or OpenLDAP. If an entry does not exist in Novell eDirectory or OpenLDAP, the entry is deleted from the Oracle back-end directory. If you assign a value of `false` to this parameter, deleted entries are synchronized according to the difference between the number of entries in the connected directory and the number of entries in the Oracle back-end directory. If the number of deleted entries is 0 or less than 0, then there are no deleted entries to synchronize. However, if the number of deleted entries is greater than 0, then the Oracle Directory Integration Platform compares each entry in the Oracle back-end directory with Novell eDirectory or OpenLDAP to identify the deleted entries to synchronize. The Oracle Directory Integration Platform continues to compare entries until it locates the same number of deleted entries as the difference between the number of entries in the connected directory and the number of entries in the Oracle back-end directory. For better performance, you should assign a value of `false` to this parameter.

Table 22–1 (Cont.) Novell eDirectory and OpenLDAP Synchronization Parameters for the Advanced Configuration Information Attribute

Parameter	Description
`ReduceFilterTimeInSeconds`	Specifies the time difference between the computer that is running the Oracle back-end directory and the computer that is running Novell eDirectory. This parameter is necessary because synchronization between the Oracle back-end directory and Novell eDirectory will not function properly if the time on the Novell eDirectory computer is earlier than the time on the Oracle back-end directory computer. You assign to this parameter a value in seconds that is equal to the time difference between the two computers. The default value is 0.
`UniqueAttribute`	Identifies the unique attribute in Novell eDirectory or OpenLDAP that can be used to search for an entry. You assign to this parameter a value of `GUID` for Novell eDirectory or `entryuuid` for OpenLDAP.
`Reconciler`	Identifies the class used by the profile for reconciliation purposes.

22.3.8 Step 8: Configure the OpenLDAP Connector to Synchronize Passwords

You cannot synchronize passwords from Novell eDirectory to the Oracle back-end directory. You can, however, synchronize passwords from OpenLDAP to the Oracle back-end directory.

Going the other direction, the Oracle Directory Integration Platform can synchronize password changes from the Oracle back-end directory to Novell eDirectory or OpenLDAP only when the directories are running SSL server-side authentication and only when the Oracle back-end directory is Oracle Internet Directory.

Two-way password synchronization is not supported from an Oracle Unified Directory back-end directory or an Oracle Directory Server Enterprise Edition back-end directory. Two-way password synchronization is only supported if the back-end directory is Oracle Internet Directory.

> **Note:** The Oracle back-end directory requires that the password be a minimum of 5 characters. If any OpenLDAP passwords are less than 5 characters, the password synchronization to the Oracle back-end directory will fail.

Perform the following tasks to synchronize passwords from OpenLDAP to the Oracle back-end directory:

1. Add a mapping rule that enables password synchronization. For example:

   ```
   userpassword: : : inetorgperson: userpassword: person
   ```

2. (Optional) This step is only required if the hashing algorithm in OpenLDAP is not compatible with the hashing algorithm in Oracle Directory Integration Platform.

> **WARNING: Completing this step will save the `userpassword` attribute in OpenLDAP as a plain-text password.**

Enable the password policy and reversible password encryption in the Oracle directory server. To do this, assign a value of `1` to the `orclPwdPolicyEnable` and `orclpwdEncryptionEnable` attributes in the entry `cn=PwdPolicyEntry,cn=common,cn=products,cn=oraclecontext,`*`DN_of_realm`*. You can do this by using `ldapmodify` and uploading an LDIF file containing the following entries:

```
dn:cn=PwdPolicyEntry,cn=common,cn=products,cn=oraclecontext,DN_of_realm.
changetype: modify
replace: orclpwdpolicyenable
orclpwdpolicyenable: 1
-
replace: orclpwdencryptionenable
orclpwdencryptionenable: 1
```

See Also:

- "Configuring the Connected Directory Connector for Synchronization in SSL Mode" on page 17-13
- The section "Configuring Mapping Rules" on page 6-3 for instructions on adding mapping rules
- If your Oracle back-end directory is Oracle Internet Directory, see the chapter on directory storage of password verifiers in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information about enabling reversible encryption.

22.3.9 Step 9: Synchronize in SSL Mode

Configure the Novell eDirectory or OpenLDAP connector for synchronization in SSL mode by following the instructions in "Configuring the Connected Directory Connector for Synchronization in SSL Mode" on page 17-13.

22.3.10 Step 10: Configure the Novell eDirectory or OpenLDAP External Authentication Plug-in

Configure the Novell eDirectory or OpenLDAP external authentication plug-in by following the instructions in "Configuring External Authentication Plug-ins" on page 17-16.

22.3.11 Step 11: Perform Post-Configuration and Administrative Tasks

Read Chapter 23, "Managing Integration with a Connected Directory" for information on post-configuration and ongoing administration tasks.

23

Managing Integration with a Connected Directory

This chapter contains information about post-configuration and ongoing administration tasks. It contains these topics:

23.1 Tasks After Configuring with a Connected Directory

Once configuration is complete, do the following:

1. Migrate data from one directory to the other as needed. This is described in "Bootstrapping Data Between Directories" on page 23-2.
2. Use the `activate` operation of the `manageSyncProfiles` command to enable the synchronization profile by entering the following command:

   ```
   manageSyncProfiles activate -profile profile_name
   ```

23.2 Typical Management of Integration with a Connected Directory

Management tasks typically include:

- Managing synchronization profiles and mapping rules:
  - Creating new profiles. You create new profiles if you need to synchronize with an additional domain controller in a multiple domain environment. You can create new profiles by using existing profiles as templates.
  - Changing configurations (attributes) in the profile.
  - Disabling profiles to allow maintenance and then reenabling them. Disabling profiles stops synchronization related to that profile.
- Managing mapping rules:
  - Creating new rules when additional attributes need to be synchronized.
  - Changing existing rules when the way attributes are synchronized needs to change.
  - Deleting or commenting out rules not required when a particular attribute is not required to be synchronized.
- Managing access control.

- Starting and stopping the Oracle directory server and the Oracle Directory Integration Platform.

This section contains these topics:

- Bootstrapping Data Between Directories
- Managing a Third-Party Directory External Authentication Plug-in

> **See Also:** *Oracle Fusion Middleware Getting Started with Oracle Identity Management* for instructions about how to use the Identity Management Grid Control Plug-in to manage integration with a connected directory.

23.2.1 Bootstrapping Data Between Directories

Bootstrapping is sometimes called data migration. To bootstrap data, perform the following steps after the third-party directory connector and plug-in configurations are complete:

1. Identify the data you want to migrate. You can choose to migrate all data in the directory or only a subset of data.
2. Use the following command to disable the import and export synchronization profile:

   ```
   manageSyncProfiles deactivate -profile profile_name
   ```

3. Bootstrap from one directory to another using the `syncProfileBootstrap` command. Refer to Chapter 8, "Bootstrapping a Directory in Oracle Directory Integration Platform" for more information about bootstrapping.

   Once bootstrapping is accomplished, the profile status attributes are appropriately updated in the synchronization profile by the `manageSyncProfiles` command.
4. If you used LDIF file-based bootstrapping, then initialize the `lastchangekey` value with `updatechgnum` operation of the `manageSyncProfiles` command as follows:

   ```
   manageSyncProfiles updatechgnum -profile profile_name
   ```

   This `lastchangekey` attribute should be set to the value of the last change number in the source directory before you started the bootstrap.
5. If two-way synchronization is required, then enable the export profile and make sure the change logging option is enabled for the Oracle directory server.

   For Oracle Internet Directory, change logging is controlled by the `-l` option while starting Oracle Internet Directory. By default, it is set to `TRUE`, meaning that change logging is enabled. If it is set to `FALSE`, then use the OID Control Utility to shut down the Oracle Internet Directory server, and then to start the server again with the change log enabled.

23.2.2 Managing a Third-Party Directory External Authentication Plug-in

This section explains how to delete, disable, and re-enable a third-party external authentication plug-in.

23.2.2.1 Deleting a Third-Party Directory External Authentication Plug-in

To delete a third-party external authentication plug-in, enter the following commands. After executing the commands, you will be prompted for a password.

```
ldapdelete -h host -p port -D binddn -q \
"cn=adwhencompare,cn=plugin,cn=subconfigsubentry"

ldapdelete -h host -p port -D binddn -q \
"cn=adwhenbind,cn=plugin,cn=subconfigsubentry"
```

23.2.2.2 Disabling a Third-Party External Authentication Plug-in

To disable a third-party external authentication plug-in:

1. Create an LDIF file with the following entries:

```
dn: cn=adwhencompare,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 0

dn: cn=adwhenbind,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 0
```

2. Load the LDIF file with the `ldapmodify` command, as follows:

```
ldapmodify -h host -p port -D binddn -q -f fileName
```

> **Note:** You will be prompted for the password.

23.2.2.3 Re-enabling a Third-Party External Authentication Plug-in

To re-enable a third-party external authentication plug-in, use these two commands:

1. Create an LDIF file with the following entries:

```
dn: cn=adwhencompare,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 1

dn: cn=adwhenbind,cn=plugin,cn=subconfigsubentry
changetype: modify
replace: orclpluginenable
orclpluginenable: 1
```

2. Load the LDIF file with the `ldapmodify` command, as follows:

```
ldapmodify -h host -p port -D binddn -q -f fileName
```

> **Note:** You will be prompted for the password.

Part VI

Appendixes

This part contains the following appendixes:

- Appendix A, "Comparing Oracle Directory Integration Platform 11g Release 1 (11.1.1) and 10g Releases (10.1.4.x)"
- Appendix B, "Example Properties File for Synchronization Profiles"
- Appendix D, "Starting and Stopping the Oracle Stack"
- Appendix C, "Case Study: A Deployment of Oracle Directory Integration Platform"
- Appendix E, "Troubleshooting the Oracle Directory Integration Platform"

A

Comparing Oracle Directory Integration Platform 11g Release 1 (11.1.1) and 10g Releases (10.1.4.x)

This appendix compares the implementation of fundamental items in Oracle Directory Integration Platform between 11*g* Release 1 (11.1.1) and legacy 10*g* Releases (10.1.4.x). The information in this appendix is provided to give you an overview of implementation changes between the releases and to provide orientation after you upgrade to 11*g* Release 1 (11.1.1).

This appendix contains the following topics:

A.1 Process Management

In 10*g* Releases (10.1.4.x):

- Start, stop, restart, and other processes were controlled using the `oidctl` command. Oracle Directory Integration Platform was a J2SE application performing synchronization and provisioning using its own scheduler.
- Multiple Oracle Directory Integration Platform server instances could be started to process different profile groups.
- The instance with configset 0 processed the provisioning profiles. The instance with different configset and groupid processed groups of synchronization profiles.

In 11*g* Release 1 (11.1.1):

- Oracle Directory Integration Platform is a J2EE application deployed on an Oracle WebLogic Managed Server with Oracle Directory Services Manager. The default name of the managed server is `wls_ods1`. Start, stop, restart, and other processes are controlled by starting and stopping the Oracle WebLogic Managed Server.

- Oracle Directory Integration Platform server is deployed and undeployed using WLST commands or the Oracle WebLogic Server Administrative (Admin) console.
- The Quartz Scheduler is used for scheduling processing.
- One instance of Oracle Directory Integration Platform schedules all synchronization and provisioning profiles.
- No concept of configset and profile groups. All enabled profiles are scheduled.

A.2 Configuration Files

In 10*g* Releases (10.1.4.x):

- The $*ORACLE_HOME*`/ldap/odi/conf/odi.properties` file contained the Oracle Directory Integration Platform server password used to connect to Oracle Internet Directory (the Oracle back-end directory). It also contained the Oracle Wallet location and password.
- The connection details for the Oracle Internet Directory associated with Oracle Directory Integration Platform were specified as part of the command to start Oracle Directory Integration Platform.
- Oracle Wallet was used for storing certificates.

In 11*g* Release 1 (11.1.1):

- The associated Oracle Internet Directory host and port details are stored in the `dip-config.xml` file in `dipapps.ear`.
- Java Keystore is used for storing the SSL certificates.
- The password Oracle Directory Integration Platform uses to connect to Oracle Internet Directory (the Oracle back-end directory) is stored in the Credential Store Framework. The JKS passwords are also stored in the Credential Store Framework.
- All parameters required for Oracle Directory Integration Platform to start are specified in the `dip-config.xml` file.

A.3 Templates for Mapping, Configuration, and Properties Files

In 10*g* Releases (10.1.4.x):

- Templates for mapping and configuration files for all connected directories were located in the $*ORACLE_HOME*`/ldap/odi/conf` directory.
- Templates for mapping, configuration, and properties files for LDIF, Tagged directories were located in the $*ORACLE_HOME*`/ldap/odi/samples` directory.
- Templates for bootstrapping files were located in the $*ORACLE_HOME*`/ldap/odi/samples` directory.

In 11*g* Release 1 (11.1.1):

- Templates for mapping and configuration files for all connected directories are located in the $*ORACLE_HOME*`/ldap/odi/conf` directory.
- Templates for mapping, configuration, and properties files for LDIF, Tagged directories are located in the $*ORACLE_HOME*`/ldap/odi/samples` directory. Refer to Appendix B, "Example Properties File for Synchronization Profiles" for more information.

- Templates for bootstrapping files are located in the $*ORACLE_HOME*/ldap/odi/samples directory. Refer to "Bootstrapping Using a Parameter File" on page 8-4 for more information.

A.4 Log Files

In 10*g* Releases (10.1.4.x):

- Oracle Directory Integration Platform server log file was located in the $*ORACLE_HOME*/ldap/log directory.
- Individual logs for each profile were located in the $*ORACLE_HOME*/ldap/odi/log/ directory. Logs used a file naming convention of *PROFILE_NAME*.log.

In 11*g* Release 1 (11.1.1):

- Log files are located at:

  $*MW_HOME*/user_projects/domains/*DOMAIN_NAME*/servers/*NAME_OF_MANAGED_SERVER*/logs/NAME_OF_MANAGED_SERVER.log

  Note: This log file contains the logs for the Oracle Directory Integration Platform server and all profiles.

A.5 Graphical User Interfaces

In 10*g* Releases (10.1.4.x):

- DIPAssistant -gui was the Graphical User Interface (GUI) tool for managing synchronization profiles.

In 11*g* Release 1 (11.1.1):

- Use Oracle Enterprise Manager Fusion Middleware Control to manage synchronization and provisioning profiles. Refer to "Using Fusion Middleware Control" on page 3-1 for more information.

A.6 Command-Line Tools

In 10*g* Releases (10.1.4.x):

- dipassistant: Was used to manage synchronization profiles.
- oidprovtool: Was used to manage provisioning profiles.

In 11*g* Release 1 (11.1.1):

- dipStatus: Allows you to check the status of Oracle Directory Integration Platform and whether or not it is registered. Refer to "Viewing the Status of Oracle Directory Integration Platform Using the dipStatus Utility" on page 4-4 for more information.
- manageDIPServerConfig: Manages Oracle Directory Integration Platform configuration settings including refresh interval, Oracle back-end directory port number, keystore location and password, and the number of scheduler threads.

Refer to "Managing Oracle Directory Integration Platform Using manageDIPServerConfig" on page 4-9 for more information.

- `manageSyncProfiles`: Manages Oracle Directory Integration Platform synchronization profiles. Refer to "Managing Synchronization Profiles Using manageSyncProfiles" on page 7-16 for more information.
- `syncProfileBootstrap`: Performs the initial migration of data between a connected target directory and the Oracle back-end directory on a synchronization profile or LDIF file. Refer to "Directory Bootstrapping Using syncProfileBootstrap" on page 8-1 for more information.
- `expressSyncSetup`: Creates profiles for standard LDAP directories using prepackaged templates based on the directory type. Refer to "Creating Import and Export Synchronization Profiles Using expressSyncSetup" on page 17-2 for more information.
- `provProfileBulkProv`: Performs initial migration of data from an LDIF file to the Oracle back-end directory for a provisioning profile. Refer to "Bulk Provisioning Using the provProfileBulkProv Tool" on page 12-7 for more information.
- `oidprovtool`: Administers provisioning profile entries in the directory by enabling you to perform tasks such as:
  - Create new provisioning profiles
  - Enable or disable existing provisioning profiles
  - Modify existing provisioning profiles
  - Delete existing provisioning profiles
  - Get the current status of a provisioning profile
  - Clear all errors in an existing provisioning profile

  Refer to "Managing Provisioning Profiles Using oidprovtool" on page 13-2 for more information.
- `schemasync`: Directory Integration Platform does not support the synchronization of schema and ACLs. You can use the `schemasync` tool to identify differences in schema, specifically attributes and object classes, between the Oracle back-end directory and connected directories. After identifying the differences, you can make the appropriate changes to the LDIF file containing the schema and then use the `ldapadd` and `ldapmodify` tools to upload the schema differences. `schemasync` is located in the *ORACLE_HOME*`/bin` directory.

  See: *Oracle Fusion Middleware Reference for Oracle Identity Management* for more information.

A.7 Audit Configurables

In 10*g* Releases (10.1.4.x):

- Audit details were available in the $*ORACLE_HOME*`/ldap/odi/log` directory. Details were maintained individually for each profile and stored in profile-specific files, such as *PROFILE_NAME*`.aud`.
- Auditing did not require any specific configuration.
- By default, audit was enabled and events were logged.

In 11*g* Release 1 (11.1.1):

- Oracle Directory Integration Platform uses the Oracle Fusion Middleware common audit framework. You can enable audit using WLST and Oracle Enterprise Manager Fusion Middleware Control.

A.8 Audit Log Location

In 10*g* Releases (10.1.4.x):

- *ORACLE_HOME*`/ldap/odi/log/`*PROFILE_NAME*`.aud`

In 11*g* Release 1 (11.1.1):

- *DOMAIN_HOME*`/servers/wls_ods1/logs/auditlogs/DIP/`

B

Example Properties File for Synchronization Profiles

This appendix provides an example of a profile properties file that can be used with the `manageSyncProfiles` command and its `-f` option. The appendix contains the following topics:

- Example Properties File for Synchronization Profiles

B.1 Example Properties File for Synchronization Profiles

The `manageSyncProfiles` command and its `-f` option allows you to specify the full path to a profile properties file that contains properties for a synchronization profile. For example:

```
manageSyncProfiles register -h myhost.mycompany.com -p 7005 -D login_ID \
  -f /opt/ldap/odip/iPlImport.profile
```

See: "Managing Synchronization Profiles Using manageSyncProfiles" on page 7-16 for more information about `manageSyncProfiles`.

The following is an example of a profile properties file. Be sure to modify your properties file so that it is specific to your environment and configuration.

```
#####################################################################
## This file contains information required to create a profile in ##
## OID.                                                            ##
#####################################################################

# Profile Name : Name of the profile
#
# NOTE - This should be a unique name
#
odip.profile.name = ActiveImport

# Profile Status : Can be either DISABLE or ENABLE
#
# NOTE - Default is DISABLE. When it is in the disable mode you can also test the
# profile using the 'testprofile' option.
#
odip.profile.status = DISABLE

# Synchronization Mode : Specifies the direction of synchronization i.e when the
# changes are required to be propagated from the 3rd party to OID then the
```

```
# synchronization mode is IMPORT. On the other hand when the changes needs ot be
# propagated to the 3rd party directory then the synchronization mode is EXPORT.
#
#
odip.profile.syncmode = IMPORT

# Retry Count : Maximum number of times this profile should be executed
# in case of an error before the integration server gives up
#
# NOTE - the default value is 4
#
odip.profile.retry = 5

# Schedule Interval: The time interval between successive execution of this
# profile by the integration server.
#
# NOTE - the default value is 60 sec. If the previous execution has not
# compeleted then the next execution will not resume util it completes.
#
odip.profile.schedinterval = 60

# Agent Execution Command : In case of a NON-LDAP interface the command
# that needs to be executed that would produce the information in LDIF/Tagged
# format. By default this property is commented out for LDAP directories.
#
odip.profile.agentexecommand =

# Connected Directory Url : The 3rd party directory location
# The property is of teh format  "host:port:sslmode"
# Host : Connected directory/repository Host
# port : connected Directory/repository Port
# sslMode: can have valid values 0,1,2,3
#  0: Non -ssl
#  1: ssl mode 1 ( no certificate )
#  2: One way SSL ) Server only Auth - Trust Point Certificate )
#
odip.profile.condirurl = host:port:sslmode

# Connected Directory/repository Account : The Dn or user name used to connect to
# the target repository
#
odip.profile.condiraccount =

# Connected Directory Account : The password used to connect to the 3rd party
# directory
# When you create a profile using the properties file you'll be prompted for the
# password even if you specify the password in this file. For security reasons it
# is recommended that you specify the password in the commandline.
# odip.profile.condirpassword = ********

# Interface Type : Whether the LDAP or LDIF or DB or TAGGED format is
# to be used for data exchange
#
# NOTE - Default value is LDAP
#
odip.profile.interface = LDAP

# Config Info : Additional information required for execution of this
# profile by the integration server.
#
```

```
# NOTE - The value for this property is the name of the file that contains
# the additional profile specific information to be used for execution
# Specify the absolute pathname of the file here. If the absolute pathname
# contains a `\`, use the escape sequence and specify it as `\\`
#
odip.profile.configfile = /scratch/americas/product/oracle/wls/Oracle_
IDM1/ldap/odi/conf/activeimp.cfg.master

# Mapping Rules : Specifies the Mapping Rules to be used for execution
# profile by the integration server.
#
# NOTE - The value for this property is the name of the file that contains
# the domain and attribute mapping rules
# Specify the absolute pathname of the file here. If the absolute pathname
# contains a `\`, use the escape sequence and specify it as `\\`
#
odip.profile.mapfile = /scratch/americas/product/oracle/wls/Oracle_
IDM1/ldap/odi/conf/activechg.map.master

# Matching filter Con Dir : Specifies the filter that needs to be
# applied to the changes that are read from the connected directory
# before importing to OID
#
# NOTE - There are certain defaults available for different directories.
# You can look at the $ORACLE_HOME/ldap/odi/conf directory for sample
# files and filters.
#
# odip.profile.condirfilter =
"searchfilter=(|(objectclass=group)(objectclass=organizationalunit)(&(objectclass=
user)(!(objectclass=computer))))"


# Matching OID attribute : Specifies the matching attribute
# on OID for import from the connected directory
#
odip.profile.oidfilter = orclObjectGUID

# Change Number : Specifies the last applied change number. In case of
# an export profile this number refer's to OID's last applied change number.
# However, in case of the import profile this number refers to the last
# applied change number in the connected directory.
#
odip.profile.lastchgnum = 0

# Profile Version : Value is 4.0. Only version 4.0 profiles are supported.
#
# NOTE - Default is 4.0
#
odip.profile.version = 4.0

# Debug Level : Specifies the debug level of the profile. A value of 63 logs all
# information, including entries that are synchronized.
#
odip.profile.debuglevel = 0

# Specify the directory type here.  Supported values are , ACTIVEDIRECTORY,ADAM,
# EDIRECTORY, IPLANET, OID, OPENLDAP, and TIVOLI
#
odip.profile.directorytype=ACTIVEDIRECTORY
```

```
# associated Profile name. Specify the profile you would like to associate with
# the current profile. This is applicable only for LDAP directories and is
# required only if you are using bi-directional
# synchronization with a connected directory. If you have only one direction of
# synchronization you can leave this field empty.
odip.profile.associatedProfile =

# updateChangeNumberatCreate: if the field is set to false,
# Last Change Number(lastchgnum) will be set to
# current time stamp or value at the time of profile creation.
# Instead you can set it true to retain its deafult value.
#
odip.profile.updateChangeNumberatCreate = TRUE
```

C

Case Study: A Deployment of Oracle Directory Integration Platform

This appendix describes a deployment in which Oracle Directory Integration Platform integrates various applications in the MyCompany enterprise.

This section contains these topics:

C.1 Components in the MyCompany Enterprise

This hypothetical enterprise has the following components:

- Oracle Human Resources, in which all employees and contractors are managed
- Oracle Internet Directory, which is the Oracle back-end directory
- Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server), a connected directory that is being used by certain applications
- Oracle Portal, which is used as the intranet portal for all employees

C.2 Requirements of the MyCompany Enterprise

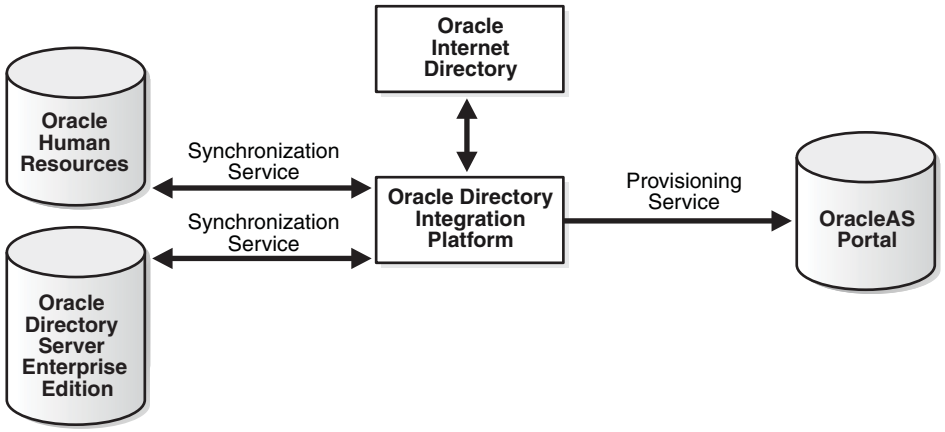
The MyCompany enterprise requires that:

- All employees and contractors are created in Oracle Human Resources. Once created, all applications in the enterprise must share this information through the Oracle back-end directory (Oracle Internet Directory).
- All applications in the enterprise, including single sign-on services, can honor any employee created in Oracle Human Resource.
- All applications that are affected by changes to user properties are notified when changes occur.
- A user's access rights are revoked when the user is terminated in Oracle Human Resources.

C.3 Overall Deployment in the MyCompany Enterprise

Figure C–1 illustrates the various components and their relationships to each other.

Figure C–1 Example of Oracle Directory Integration Platform in the MyCompany Deployment



In the example in Figure C–1:

- Oracle Internet Directory (the Oracle back-end directory) is the central user repository for all enterprise applications.
- Oracle Human Resources is the basis for all user-related information. It is synchronized with Oracle Internet Directory by using the Oracle Directory Synchronization Service.
- Oracle Directory Server Enterprise Edition, which is already deployed in the enterprise, is synchronized with Oracle Internet Directory by using the Oracle Directory Synchronization Service.
- Oracle Portal is notified of changes in Oracle Internet Directory by using the Oracle Directory Integration Platform Service.

C.4 User Creation and Provisioning in the MyCompany Enterprise

In this example, the MyCompany enterprise requires that all users be created in Oracle Human Resources. Oracle Directory Integration Platform must propagate new user records to all other repositories in the enterprise.

Figure C–2 illustrates how Oracle Directory Integration Platform performs this task.

Figure C–2 User Creation and Provisioning

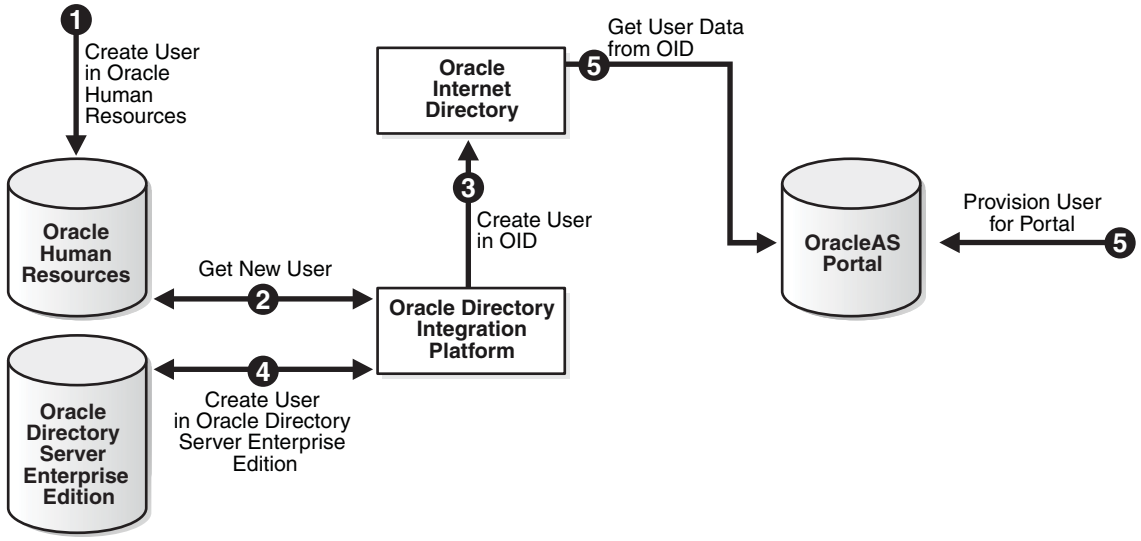


Figure C–2 shows the creation of a new user in Oracle Human Resources, which, in turn, causes an entry for that user to be created in Oracle Internet Directory and Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server). It also shows the process of provisioning the user to access the Oracle Portal application. User creation and provisioning occur in the following manner:

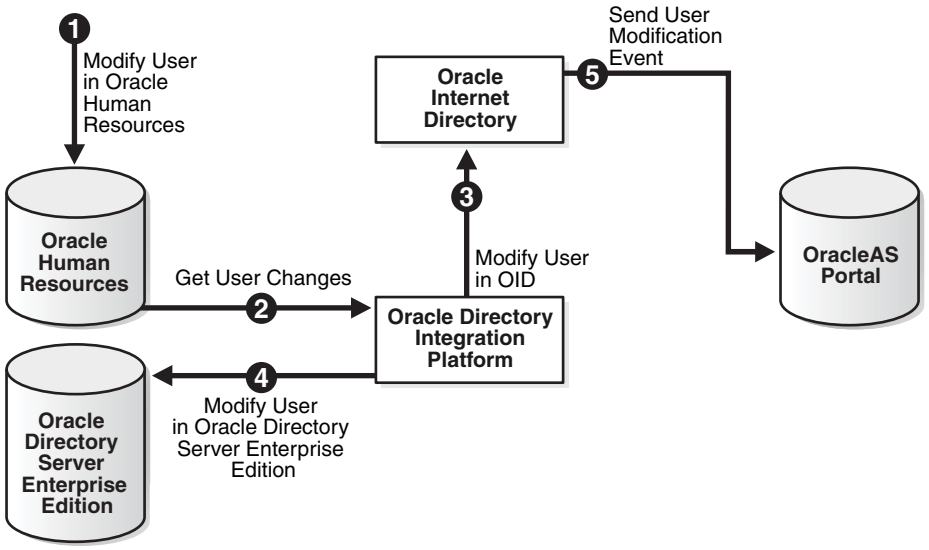
1. The Oracle Human Resources administrator creates the user in the Oracle Human Resources database.
2. Oracle Directory Integration Platform, through the Oracle Directory Synchronization Service, detects the new-user creation.
3. Oracle Directory Integration Platform, through the Oracle Directory Synchronization Service creates the entry for the user in Oracle Internet Directory.
4. Oracle Directory Integration Platform, through the Oracle Directory Synchronization Service, creates an entry in the Oracle Directory Server Enterprise Edition.
5. Because the user entry is available in Oracle Internet Directory, the Oracle Portal administrator can now provision the user to use the services of Oracle Portal. During this task, the Oracle Portal software automatically retrieves the user information from Oracle Internet Directory.

Note that Oracle Directory Integration Platform does not directly notify Oracle Portal about new users. This is because not all users created in Oracle Human Resources need access to all services. In this case, the deployment must explicitly provision the users to use these services, as in Step 5.

C.5 Modification of User Properties in the MyCompany Enterprise

In this example, the MyCompany enterprise requires that any modification to user properties be communicated to all components interested in such changes. Figure C–3 illustrates the actions that Oracle Directory Integration Platform takes to meet this requirement.

Figure C–3 Modification of User Properties



The process is as follows:

1. The user is first modified in Oracle Human Resources.
2. Oracle Directory Integration Platform retrieves these changes through the Oracle Directory Synchronization Service.
3. Oracle Directory Integration Platform makes the corresponding user modification in Oracle Internet Directory.
4. The Oracle Directory Synchronization Service modifies the user in Oracle Directory Server Enterprise Edition.
5. Oracle Directory Integration Platform, through the Oracle Directory Integration Platform Service, notifies Oracle Portal about the change in user properties.

C.6 Deletion of Users in the MyCompany Enterprise

In this example, the MyCompany enterprise requires that a user being deleted or terminated in Oracle Human Resources be automatically denied access to all enterprise resources that are based on the directory service.

Figure C–4 shows the flow of events during the deletion of users.

Figure C–4 Deletion of Users from the Corporate Human Resources

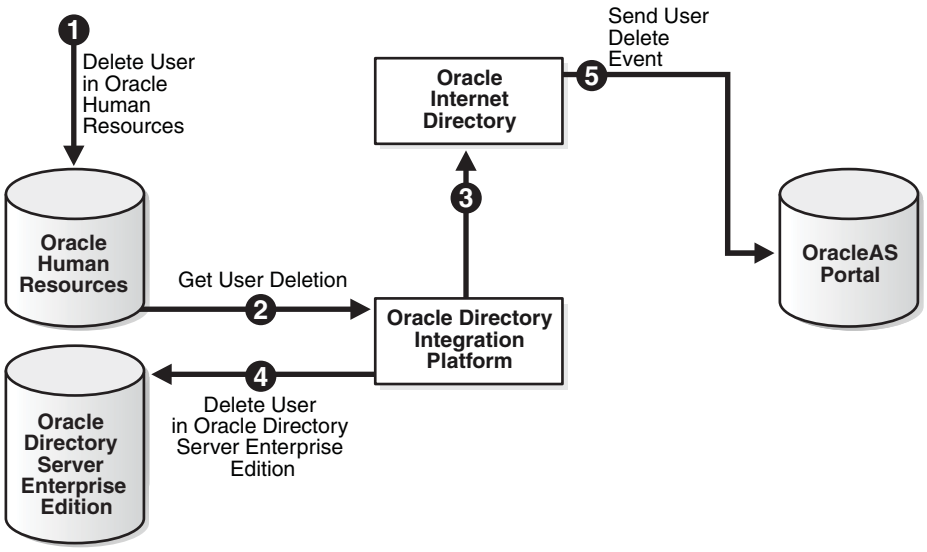


Figure C–4 shows the process by which Oracle Directory Integration Platform communicates the deletion of users to all systems in the enterprise. The process is as follows:

1. The user is first deleted in Oracle Human Resources.
2. Oracle Directory Integration Platform retrieves these changes through the Oracle Directory Synchronization Service.
3. Oracle Directory Integration Platform, through the Oracle Directory Synchronization Service, makes the corresponding user deletion in Oracle Internet Directory.
4. Oracle Directory Integration Platform, through the Oracle Directory Synchronization Service, deletes the users in Oracle Directory Server Enterprise Edition.
5. Oracle Directory Integration Platform, through the Oracle Directory Integration Platform Service, notifies Oracle Portal about the deletion of the user.

Once all of the steps are completed, a deleted user in Oracle Human Resources cannot access Oracle Portal.

D

Starting and Stopping the Oracle Stack

You must start and stop the components of the Oracle stack in a specific order, which is described in this appendix.

Starting the Stack

Start the stack components in the following order.

1. Start the Oracle Database
   a. In the Database `ORACLE_HOME`, set the `ORACLE_SID`, `ORACLE_HOME` and `PATH` environment variables to the appropriate values.
   b. Start the listener.

      ```
      ORACLE_HOME/bin/lsnrctl start
      ```

   c. Start the database.

      ```
      ORACLE_HOME/bin/sqlplus "/as sysdba"
      startup
      ```

2. Start the Oracle WebLogic Administration Server.

   Note: If you start the Oracle WebLogic Administration Server from the command line, it runs in the foreground and prints output to the screen.

   ```
   MW_HOME/user_projects/domains/DOMAIN_NAME/bin/startWebLogic.sh \
   SERVER_NAME {ADMIN_URL}
   ```

   When executing these scripts:

   - The default value for *DOMAIN_NAME* is `IDMDomain`
   - *SERVER_NAME* represents the name of the Oracle WebLogic Managed Server. Its default value is `wls_ods1`.
   - You will be prompted for values for *USER_NAME* and *PASSWORD* if you do not provide them as options when you execute the script.
   - The value for *ADMIN_URL* will be inherited if you do not provide it as an option when you execute the script.

3. Ensure that the Node Manager is running. Normally, the Oracle WebLogic Administration Server starts the Node Manager. If, for some reason, the Node Manager is not running, start it.

```
MW_HOME/user_projects/domains/DOMAIN_NAME/bin/startNodeManager.sh
```

4. Start system components, such as Oracle Internet Directory.

   ```
   ORACLE_INSTANCE/bin/opmnctl startall
   ```

   You can verify that the system components have started by executing:

   ```
   ORACLE_INSTANCE/bin/opmnctl status -l
   ```

5. Start WebLogic managed components, such as Oracle Directory Integration Platform and Oracle Directory Services Manager.

   ```
   MW_HOME/user_projects/domains/DOMAIN_NAME/bin/startManagedWebLogic.sh \
     SERVER_NAME {ADMIN_URL}
   ```

   To start managed components in the background, you can use the Oracle WebLogic Administration Console. See *Oracle Fusion Middleware Introduction to Oracle WebLogic Server* for more information.

   You can view the status of WebLogic managed components with Oracle Enterprise Manager Fusion Middleware Control.

Stopping the Stack

You can stop the Administration Server and all the managed servers by using Oracle WebLogic Administration Console. See *Oracle Fusion Middleware Introduction to Oracle WebLogic Server* for more information.

To stop the stack components from the command line, issue the commands in the following order.

1. Stop WebLogic managed components, such as Oracle Directory Integration Platform and Oracle Directory Services Manager.

   ```
   MW_HOME/user_projects/domains/DOMAIN_NAME/bin/stopManagedWebLogic.sh \
   {SERVER_NAME} {ADMIN_URL} {USER_NAME} {PASSWORD}
   ```

2. Stop system components, such as Oracle Internet Directory.

   ```
   ORACLE_INSTANCE/bin/opmnctl stopall
   ```

3. Stop the WebLogic Administration Server.

   ```
   MW_HOME/user_projects/domains/DOMAIN_NAME/bin/stopWebLogic.sh
   ```

4. If you want to stop the Node Manager, you can use the `kill` command.

   ```
   kill -9 pid
   ```

5. Stop the Oracle Database.

   a. In the Database `ORACLE_HOME`, set the `ORACLE_SID`, `ORACLE_HOME`, and `PATH` environment variables to the appropriate values.

   b. Stop the database.

      ```
      ORACLE_HOME/bin/sqlplus "/as sysdba"
      shutdown immediate
      ```

   c. Stop the listener.

      ```
      ORACLE_HOME/bin/lsnrctl stop
      ```

E

Troubleshooting the Oracle Directory Integration Platform

This appendix describes common problems that you might encounter when using the Oracle Directory Integration Platform and explains how to solve them. It contains these topics:

See Also:

- *Oracle by Example for Oracle Identity Management*, available from the Oracle Technology Network at `http://www.oracle.com/technology/index.html`
- *Oracle Identity Management User Reference*

E.1 Checklist for Troubleshooting Oracle Directory Integration Platform

Use the following checklist as a starting point when troubleshooting Oracle Directory Integration Platform problems:

- Verify that the Oracle Directory Integration Platform application has been deployed by using the WebLogic console.
- Verify that the Oracle Directory Integration Platform application is running.

  To verify the status of the Oracle Directory Integration Platform application using Oracle Enterprise Manager Fusion Middleware Control, open a Web browser and enter the Oracle Enterprise Manager Fusion Middleware Control URL for your environment. The format of the Oracle Enterprise Manager Fusion Middleware Control URL is: `https://`*host*`:`*port*`/em`.

  You can view the status of the Oracle Directory Integration Platform application in the status column of the Fusion Middleware section on the Oracle Enterprise Manager Fusion Middleware Control home page for your environment.

  To verify the status of the Oracle Directory Integration Platform application from the command-line, use the `dipStatus` utility. If Oracle Directory Integration Platform is running, `dipStatus` returns an `ODIP Application is active at this host and port` message.

Notes:

- When using `dipStatus`, be sure you specify the host and port of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed, not the host and port of the Administration Server.
- Refer to "Viewing the Status of Oracle Directory Integration Platform Using the dipStatus Utility" on page 4-4 for more information.

- Verify the appropriate profiles are enabled by listing their names and status using the `manageSyncProfiles` command as follows:

```
manageSyncProfiles list -h host -p port -D user [-prfSt] [-help]
```

Note: You will be prompted for the password.

- Verify that the third-party LDAP directory server is running by executing the following command:

```
ldapbind -h ldap_host -p ldap_port -D binddn -q
```

Note: You will be prompted for the password.

- If you are using the PL/SQL plug-in, use `sqlplus` to verify that you can connect to the provisioning-integrated application.

E.2 Problems and Solutions

This section describes common problems and solutions for Oracle Directory Integration Platform. It contains these topics:

- Provisioning Errors and Problems
- Synchronization Errors and Problems
- Windows Native Authentication Errors and Problems
- Novell eDirectory and OpenLDAP Synchronization Errors and Problems
- Oracle Password Filter for Microsoft Active Directory Errors and Problems

Note: The Oracle Directory Integration Platform stores error messages in the appropriate file, as described in "Location and Naming of Files" on page 6-20.

E.2.1 Provisioning Errors and Problems

This section provides solutions for provisioning errors and problems.

Problem

Unable to get the Entry from its GUID. Fatal Error...

Solution

Oracle Directory Integration Platform is attempting to retrieve an entry that has been deleted, but appears to not have been purged. However, when this error happens, the entry has been already purged. To avoid future errors, update the tombstone purge configuration settings in the Oracle Internet Directory garbage collection framework by referring to the "Managing Garbage Collection" chapter in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

Problem

LDAP connection failure.

Solution

Oracle Directory Integration Platform failed to connect to the directory server. Check the connection to the directory server.

> **See Also:** If your Oracle back-end directory is Oracle Internet Directory, see the chapter about directory server administration in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*. This chapter contains information about directory server connections.

Problem

Initialization and database connection failures, and exceptions while calling an SQL operation.

Solution

To test the connection, use the **Test Connection** feature for the profile in Oracle Enterprise Manager Fusion Middleware Control. If the connection fails, examine the diagnostic log file at the following location for more information:

MW_HOME/user_projects/domains/*DOMAIN_NAME*/servers/*NAME_OF_MANAGED_SERVER*/logs/

Note: The file name is *NAME_OF_MANAGED_SERVER*-diagnostic.log

---

Problem

Provisioning Profiles Not Getting Executed by the DIP Provisioning Server.

Solution

Using Oracle Enterprise Manager Fusion Middleware Control or the `oidprovtool` command, verify the profile is enabled and that the Oracle Directory Integration Platform scheduling interval is set to a positive integer.

Problem

Unable to Connect to the Application Database.

Solution

The application database connection requirements in a provisioning profile may be incorrect. Use `sqlplus` to verify connectivity requirements.

Problem

User/Group Modify And Delete Events Not being consumed by the application.

Solution

Verify the host port details and credentials using the **Test Connection** feature for the profile in Oracle Enterprise Manager Fusion Middleware Control. If the connection fails after using the **Test Connection** option, an error message appears providing information about the failed connection.

For additional information about the failed connection, you can examine the diagnostic log using Oracle Enterprise Manager Fusion Middleware Control or from the command line. The diagnostic log is located at:

MW_HOME/user_projects/domains/*DOMAIN_NAME*/servers/*NAME_OF_MANAGED_SERVER*/logs/

> **Note:** The file name is *NAME_OF_MANAGED_SERVER*-diagnostic.log

Problem

Subscription to binary attributes results in the event propagation error.

Solution

Binary attributes propagation is not supported. Remove the binary attribute assignments from the event subscription in the provisioning profile.

Problem

Insufficient Access Rights to do "proxy" as the Application DN.

Solution

The Oracle Directory Integration Platform server group has not been granted browse privilege by the application DN. Use the `ldapmodify` command to load the following ACIs, which grant browse privileges from the application DN to the Oracle Directory Integration Platform group:

```
orclaci: access to attr=(*) by group="cn=odisgroup,cn=DIPAdmins,cn=Directory
Integration Platform,cn=products,cn=oraclecontext "(read,write,search,compare)
orclaci: access to entry by group="cn=odisgroup,cn=DIPAdmins,cn=Directory
Integration Platform,cn=products,cn=oraclecontext"(browse,proxy)
```

Problem

Insufficient access rights to use an application DN as a proxy.

Solution

The Oracle Directory Integration Platform server group has not been granted proxy privileges by the application DN. Use the `ldapmodify` command to load the following ACI, which grants proxy privileges from the application DN to the Oracle Directory Integration Platform group:

```
orclaci: access to entry by group=" cn=odisgroup, cn=odi,cn=oracle internet
directory" (browse,proxy)
```

E.2.2 Synchronization Errors and Problems

This section provides solutions for synchronization errors and problems.

> **See Also:** Note: 276481.1—*Troubleshooting OID DIP Synchronization Issues* in My Oracle Support (formerly MetaLink) at http://metalink.oracle.com/

Problem

```
LDAP: error code 50 - Insufficient Access Rights; remaining name
'CN=Users,dc=mycompany,dc=com'
```

Solution

The record target is not in a default container. Find the `DST CHANGE RECORD`. Check the ACIs for the target container. If they are blank, then use DIP Tester to apply a known set of ACIs to the new container.

Problem

LDAP: error code 50 - Insufficient Access Rights; ACTIVECHGIMP MAPPING IMPORT OPERATION FAILURE; Agent execution successful, Mapping/import operation failure

Solution

By default the `cn=Users,`*`default realm`* contains the proper ACIs. However, this error can occur when trying to synchronize into a different container within the default realm. Open the trace file, locate the change record that is causing the error, and then check the ACIs for the record's parent container. Apply the same ACIs to the target container.

Problem

```
Log File Error: Not able to construct DN Output ChangeRecord :
Changetype:   1 ChangeKey: cn=users, dc=us,dc=oracle,dc=com
Exception  javax.naming. ContextNotEmptyException: [LDAP: error
code 66 - Not Allowed On   Non-leaf]; remaining name
'cn=users,dc=us,dc=oracle,dc=com' Missing mandatory
attribute(s).
```

Solution

There is a problem with the mapping file. Refer to Note: 261342.1—*Understanding DIP Mapping* in My Oracle Support (formerly MetaLink) at http://metalink.oracle.com/.

Problem

```
Trace File Error: IPlanetImport:Error in Mapping
Enginejava.lang.NullPointerException
java.lang.NullPointerException  at
oracle.ldap.odip.engine.Connector.setValues(Connector.java:101).
```

Solution

The `orclcondirlastappliedchgnum` attribute is null or has no value. This may occur if bootstrapping failed or if you manually populated the Oracle back-end directory and did not assign a value to the `orclcondirlastappliedchgnum` attribute. Verify that the `orclcondirlastappliedchgnum` attribute has a value. If it

does not have a value, set it using the `DIP Tester` utility or using WLST to configure the DIP Mbean.

Problem

Add and change operations are successful, but delete operations fail without being recorded in the trace file.

Solution 1

Tombstones are not enabled in Oracle Directory Server Enterprise Edition or Sun Java System Directory Server. Verify that tombstones are enabled by referring to Note: 219835.1 in My Oracle Support (formerly MetaLink) at `http://metalink.oracle.com/`.

Solution 2

In Microsoft Active Directory, the account used for the profile is not a member of the DIR SYNCH ADMIN group. This only occurs if you are not using a Microsoft Active Directory administrator account. Install the appropriate patch from Microsoft.

Problem

Data synchronization problems encountered after configuring Oracle Directory Integration import or export connectors to third-party LDAP directories.

Solution

Determine the cause using the `testProfile` operation of the `manageSyncProfiles` command.

Problem

Editing the attribute mapping rule for a synchronization profile using Oracle Enterprise Manager Fusion Middleware Control may cause the `Schema not initialized for object class` error.

Solution

The problem could be caused by an invalid directory type specified for the third party directory connection details. Verify you have specified the correct directory type and connection details.

Problem

The Oracle back-end directory profile in Oracle Enterprise Manager Fusion Middleware Control shows "synchronization successful" yet no changes show up in the directory.

Solution

First, determine if synchronization is occurring by examining the following parameters for the synchronization profile using Oracle Enterprise Manager Fusion Middleware Control:

- Successful Completion Time (on DIP Server Home page)
- Last Execution Time (on DIP Server Home page)
- Scheduling Interval (on Advanced tab for profile)

Synchronization *is* occurring if the Successful Completion Time and Last Execution Time metrics have time values relevant to the current time of the system. If these

metrics indicate time values that are considerably older than the current time of the system, synchronization is *not* occurring.

If synchronization *is* occurring:

- Verify synchronization is configured to occur in the correct location by examining the **Source Container** setting on the profile's Mapping tab in Oracle Enterprise Manager Fusion Middleware Control.
- Verify the correct objects are being filtered by examining the **Source Matching Filter** setting on the profile's Filtering tab in Oracle Enterprise Manager Fusion Middleware Control.

If synchronization *is not* occurring:

- Verify the synchronization profile is enabled using the DIP Server Home page in Oracle Enterprise Manager Fusion Middleware Control.
- Check the status of the Quartz Scheduler using the DIP Server Home page in Oracle Enterprise Manager Fusion Middleware Control.
- Test the synchronization profile using the `manageSyncProfiles` command and its `testProfile` operation. Refer to "Managing Synchronization Profiles Using manageSyncProfiles" on page 7-16 for more information about the `manageSyncProfiles` command.

E.2.3 Windows Native Authentication Errors and Problems

This section provides solutions for errors and problems you may encounter when integrating Oracle Identity Management with Windows Native Authentication.

Note: Oracle Directory Integration Platform 11*g* Release 1 (11.1.1) interoperates with and supports Oracle Application Server Single Sign-On 10*g* Release 10.1.4.3.0 and higher.

See Also: The "Problems and Solutions for Windows Native Authentication Errors" section in the Troubleshooting chapter of the *Oracle Enterprise Single Sign-On Suite Plus Administrator's Guide* for more information about Windows Native Authentication errors.

Problem

Internal Server error. Please contact your administrator.

Solution

Windows Native Authentication is misconfigured on the middle-tier computer. To fix this problem, perform the following steps:

1. Check the `opmn.log` file for errors.
2. Check the `ssoServer.log` file for errors.
3. Make sure that the keytab file is located in the `$ORACLE_HOME/j2ee/OC4J_SECURITY/config` directory and that the principal name configured in the `jazn-data.xml` file is correct.
4. Make sure that the single sign-on middle tier computer is properly configured to access the Key Distribution Center.

Problem

Could not authenticate to KDC.

Solution

This error message may be invoked if the realm name in `krb5.conf` is incorrectly configured. Check the values `default_realm` and `domain_realm` in `/etc/krb5/krb5.conf`. Note that the realm name is case-sensitive.

Problem

Your browser does not support the Windows Kerberos authentication or is not configured properly.

Solution

The user's Web browser is not supported or is misconfigured. Follow the instructions in "Task 2: Configure Internet Explorer for Windows Native Authentication" on page 18-12.

Problem

"Access forbidden" or "HTTP error code 403" or "Windows Native Authentication Failed. Please contact your administrator."

Solution

These error messages have the same cause: the user entry cannot be found in the Oracle back-end directory. A local administrator working at a Windows desktop may be trying to access a single sign-on partner application whose entry may not have been synchronized with the Oracle back-end directory. Determine whether the user entry exists in the directory and if the Kerberos principal attributes for the user are properly synchronized from Microsoft Active Directory.

Problem

The Windows login dialog box (with user name, password, and domain fields in it) comes up when accessing the partner application.

Solution

The single sign-on server was not able to authenticate the Kerberos token because the corresponding user entry could not be found in the Oracle back-end directory. Add the user entry to the directory.

Problem

Single sign-on server fails to start. Log file contains an exception bearing the message "Credential not found."

Solution

The parameter `kerberos-servicename` may not be configured correctly. To fix this problem, perform the following steps:

1. Make sure that `kerberos-servicename` is configured correctly in the files `orion-application.xml` and `jazn-data.xml`. In the `orion-application.xml` file, the format for this parameter is `HTTP@sso.mycompany.com`. In the `jazn-data.xml` file, the format is `HTTP/sso.mycompany.com`.
2. Check the `ssoServer.log` file for errors.

3. Make sure that the keytab file is located in the $*ORACLE_HOME*`/j2ee/OC4J_SECURITY/config` directory and that the principal name configured in `jazn-data.xml` is correct.
4. Make sure that the single sign-on middle tier computer is configured to access the Kerberos domain controller.

Problem

The following exception is raised when running the OracleAS Single Sign-On Server Configuring Assistant:

```
Repository Access API throws exception :
oracle.ias.repository.schema.SchemaException: Unable to establish secure
connection to Oracle Internet Directory Server
ldap://server.mycompany.com:636/ Base Exception :
javax.naming.CommunicationException: server.mycompany.com:636 [Root
exception is java.lang.UnsatisfiedLinkError: no njssl10 in java.library.path]
        at
oracle.ias.repository.directory.DirectoryReader.connectSsl(DirectoryReader.java:
98)
        at
oracle.ias.repository.directory.DirectoryReader.connect(DirectoryReader.java:106
)
        at oracle.ias.repository.IASSchema.getDBPassword(IASSchema.java:440)
        at
oracle.ias.repository.SchemaManager.getDBPassword(SchemaManager.java:310)
        at oracle.security.sso.IMWNAConfig.getSSOHost(IMWNAConfig.java:903)
        at oracle.security.sso.IMWNAConfig.parseArgs(IMWNAConfig.java:168)
        at oracle.security.sso.IMWNAConfig.init(IMWNAConfig.java:194)
        at oracle.security.sso.IMWNAConfig.work(IMWNAConfig.java:60)
        at
oracle.security.sso.SSOConfigAssistant.wnaConfig(SSOConfigAssistant.java:243)
        at
oracle.security.sso.SSOConfigAssistant.main(SSOConfigAssistant.java:218)
```

Solution

This exception occurs when the Windows version of the OracleAS Single Sign-On Server Configuring Assistant is run on UNIX and Linux platforms. Run the UNIX/Linux version of the OracleAS Single Sign-On Server Configuring Assistant by following the instructions in "Run the OracleAS Single Sign-On Server Configuration Assistant on each Oracle Application Server Single Sign-On Host" on page 18-11.

Problem

With Windows Native Authentication, Internet Explorer is sending NT Lan Manager (NTLM) authentication instead of Kerberos credentials.

Solution

This issue is caused by an improperly configured Microsoft Active Directory installation. Refer to your Microsoft Active Directory documentation or contact Microsoft for information on how to resolve this issue.

Problem

Individual users cannot log in from specific computers using Windows Native Authentication.

Solution

If the users can log in using another computer, then there is a configuration problem with Windows or Internet Explorer on the original computer. Refer to the Microsoft Developer Network at http://msdn.microsoft.com or contact Microsoft for information on how to resolve this issue.

E.2.4 Novell eDirectory and OpenLDAP Synchronization Errors and Problems

This section provides solutions to synchronization errors and problems that can occur with Novell eDirectory and OpenLDAP.

Problem

After configuring import synchronization, entries are not synchronizing from Novell eDirectory or OpenLDAP to the Oracle back-end directory, even though the profile's synchronization status is successful and the trace file does not show any exceptions.

Possible causes and their solutions:

Cause Incorrect value assigned to the `modifiersname` parameter of the `odip.profile.condirfilter` property in the import profile.

Solution Copy the connection DN from the Novell eDirectory or OpenLDAP export profile to the `modifiersname` parameter of the `odip.profile.condirfilter` property in the import profile.

Cause The entries that the Oracle Directory Integration Platform are attempting to synchronize are created using the same DN that is assigned to the `modifiersname` parameter of the `odip.profile.condirfilter` property in the import profile.

Solution Change the DN that is assigned to the `modifiersname` parameter of the `odip.profile.condirfilter` property in the import profile to a DN that does not create the entries in Novell eDirectory of OpenLDAP.

Cause There is a time difference between the computer that is running the Oracle back-end directory and the computer that is running Novell eDirectory or OpenLDAP.

Solution Assign to the `ReduceFilterTimeInSeconds` parameter of the `odip.profile.configfile` property in the import profile a value in seconds that is equal to the time difference between the two computers.

Problem

Unsupported exception thrown during reconciliation.

Solution

One or more of the Oracle back-end directory attributes that are specified in the Novell eDirectory or OpenLDAP reconciliation rules are not indexed. Index the corresponding attributes in the Oracle back-end directory.

Problem

Deleted entries are not synchronizing from Novell eDirectory or OpenLDAP to the Oracle back-end directory, even though the profile's reconciliation status is successful.

Possible causes and their solutions:

Cause The deleted entries are not specified in the Novell eDirectory or OpenLDAP reconciliation rules.

Solution Modify the Novell eDirectory or OpenLDAP reconciliation rules to include the deleted entries.

Cause There are more entries in Novell eDirectory or OpenLDAP for a particular reconciliation rule than there are in the Oracle back-end directory.

Solution Examine the $*ORACLE_HOME*/ldap/odi/log/*profile_name*.trc file for the following message:

```
No. of entries are less in destination directory compared to source directory.
```

The preceding message is usually generated when the entire Novell eDirectory or OpenLDAP DIT needs to be synchronized with the Oracle back-end directory. To resolve this problem, assign a value of `true` to the `CheckAllEntries` parameter of the `odip.profile.configfile` property.

> **Caution:** Assigning a value of `true` to the `CheckAllEntries` parameter of the `odip.profile.configfile` property will result in decreased performance.

E.2.5 Oracle Password Filter for Microsoft Active Directory Errors and Problems

This section provides solutions to errors and problems that can occur with the Oracle Password Filter for Microsoft Active Directory.

Problem

The Oracle Password Filter for Microsoft Active Directory cannot be installed and the following error is reported in the log:

```
(Aug 23, 2010 8:26:52 PM), Install,
com.oracle.installshield.adpwd.ldapModify, dbg, C:\Program Files
(x86)\oracle\ADPasswordFilter\prepAD.ldif

(Aug 23, 2010 8:26:52 PM), Install,
com.oracle.installshield.adpwd.ldapModify, err, in LDAPOperation

(Aug 23, 2010 8:26:52 PM), Install,
com.oracle.installshield.adpwd.ldapModify, err, [LDAP: error
code 19 - 000020B5: AtrErr: DSID-03152704, #1:

0: 000020B5: DSID-03152704, problem 1005 (CONSTRAINT_ATT_TYPE),
data 0, Att 9030e (objectCategory)
```

Cause

This error may occur if the ActiveDirectory `schemaNamingContext` object does not come under the `defaultNamingContext`.

Solution

To solve this problem do one of the following:

- Replace the `ObjectCategory` attribute in `prepAD.ldif` with the value of `CN=Organizational-Unit,` *schemaNaming Context*
  where schemaNamingContext is replaced by the schema naming context value.

- Remove the `ObjectCategory` attribute from `prepAD.ldif`. Because the entry gets added in ActiveDirectory, the `objectcategory` attribute will be populated with the right value automatically.

Problem

Unable to find log file path.

Cause

Invalid log file path.

Solution

Specify a valid log file path by following the instructions in "Reconfiguring the Oracle Password Filter for Microsoft Active Directory" on page 19-16.

Problem

Cannot connect to Oracle Internet Directory in non-SSL mode (mode 1).

> **Note:** Oracle Unified Directory and Oracle Directory Server Enterprise Edition do not support non-SSL mode (mode 1).

Cause

Invalid Oracle Internet Directory configuration settings.

Solution

Correct the Oracle Internet Directory configuring settings by following the instructions in "Reconfiguring the Oracle Password Filter for Microsoft Active Directory" on page 19-16.

Problem

Cannot connect to the Oracle back-end directory in SSL mode.

Cause

The Oracle back-end directory certificate authority's trusted certificate has not been imported into the Microsoft Active Directory domain controller.

Solution

Import the trusted certificate into Microsoft Active Directory by following the instructions in "Importing a Trusted Certificate into a Microsoft Active Directory Domain Controller" on page 19-5.

Problem

Cannot connect to Microsoft Active Directory.

Cause

Invalid Microsoft Active Directory configuration settings.

Solution

Correct the Microsoft Active Directory configuration settings by following the instructions in "Reconfiguring the Oracle Password Filter for Microsoft Active Directory" on page 19-16.

Problem

Cannot upload the `prepAD.ldif` file.

Cause

The specified Microsoft Active Directory base DN container cannot store `organizationalUnit` objects.

Solution

Specify a base DN for Microsoft Active Directory that can store `organizationalUnit` objects by following the instructions in "Reconfiguring the Oracle Password Filter for Microsoft Active Directory" on page 19-16.

Problem

Password updates are looping between the Oracle back-end directory and Microsoft Active Directory.

Cause

The Oracle Password Filter is not configured to use the same bind DN and password that are specified in the synchronization profile that imports values from Microsoft Active Directory into the Oracle back-end directory.

Solution

Configure the Oracle Password Filter to use the same bind DN and password that are specified in the synchronization profile that imports values from Microsoft Active Directory into the Oracle back-end directory by following the instructions in "Reconfiguring the Oracle Password Filter for Microsoft Active Directory" on page 19-16.

Problem

Some passwords are not synchronizing between the Oracle back-end directory and Microsoft Active Directory.

Cause

The Oracle back-end directory and Microsoft Active Directory specify conflicting password policies.

Solution

Set the Oracle back-end directory password policies to the same policies that are set in Microsoft Active Directory or remove the password policies from the Oracle back-end directory.

Problem

Passwords are not synchronizing for some users.

Cause

You performed an advanced installation of the Oracle Password Filter and specified different values for the attributes that you want to synchronize between the Oracle back-end directory and Microsoft Active Directory.

Solution

Specify the same values for the attributes that you want to synchronize between the Oracle back-end directory and Microsoft Active Directory by following the instructions in "Reconfiguring the Oracle Password Filter for Microsoft Active Directory" on page 19-16.

Problem

User data synchronizes, but password synchronization is delayed.

Cause

Different time intervals are specified for user data synchronization and password synchronization.

Solution

Verify that the value assigned to the Oracle Password Filter's `SleepTime` parameter is the same as the default scheduling interval for the synchronization profile. You can use Oracle Enterprise Manager Fusion Middleware Control tool or the `manageSyncProfiles` command to view and change the default scheduling interval for synchronization profiles. To change the value assigned to the `SleepTime` parameter, follow the instructions in "Reconfiguring the Oracle Password Filter for Microsoft Active Directory" on page 19-16.

E.3 Troubleshooting Synchronization

This section describes how to troubleshoot synchronization with Oracle Directory Integration Platform. It contains these topics:

- Oracle Directory Integration Platform Synchronization Process Flow
- Understanding Synchronization Profile Registration
- Understanding the diagnostic.log File

E.3.1 Oracle Directory Integration Platform Synchronization Process Flow

When debugging synchronization issues between the Oracle back-end directory and a connected directory, it helps to understand the synchronization process flow of the Oracle Directory Integration Platform.

E.3.1.1 Oracle Directory Integration Platform Synchronization Process Flow for an Import Profile

The Oracle Directory Integration Platform reads all import profiles at startup. For each profile that is set to `ENABLE`, the Oracle Directory Integration Platform performs the following tasks during the synchronization process:

1. Connects to a third-party directory.
2. Gets the value of the last change key from the connected directory.
3. Connects to the Oracle back-end directory.
4. Gets the value of the profile's last applied change key from the Oracle back-end directory.
5. If connecting from the Oracle back-end directory to Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server), the Oracle Directory Integration Platform searches the remote change logs for entries greater

than the value of the last applied change key and less than or equal to the value of the last change key. For Microsoft Active Directory connections, the Oracle Directory Integration Platform searches for this information in the remote directory's `USNChanged` values. For the Novell eDirectory and OpenLDAP connectors, changes are identified based on the `modifytimestamp` attribute of each entry. For other types of connectors, such as the Oracle Human Resources connector, the Oracle Directory Integration Platform performs similar types of searches, although the method by which data is exchanged varies according to the type of connection.

6. Maps the data values from the connected directory to the Oracle back-end directory values.
7. Creates an Oracle back-end directory change record.
8. Applies the change (add, change, delete) in the Oracle back-end directory.
9. Updates the Oracle back-end directory import profile with the last execution times and the last applied change key from the connected directory.
10. Enters sleep mode for the number of seconds specified for the synchronization interval.

E.3.1.2 Oracle Directory Integration Platform Synchronization Process Flow for an Export Profile

The Oracle Directory Integration Platform reads all export profiles at startup. For each profile that is set to `ENABLE`, the Oracle Directory Integration Platform performs the following tasks during the synchronization process:

1. Connects to a third-party directory.
2. Connects to the Oracle back-end directory.
3. Gets the value for the last change key from the Oracle back-end directory.
4. Gets the value of the profile's last applied change key from the Oracle back-end directory.
5. The Oracle Directory Integration Platform searches the Oracle back-end directory change logs for entries greater than the value of the last applied change key and less than or equal to the value of the last change key.
6. Maps the data values from the Oracle back-end directory to the connected directory values.
7. Creates a change record.
8. Applies the change (add, change, delete) on the connected directory.
9. Updates the Oracle back-end directory export profile with the last execution times and the last applied change key from the Oracle back-end directory.
10. Enters sleep mode for the number of seconds specified for the synchronization interval.

E.3.2 Understanding Synchronization Profile Registration

This section provides information about synchronization profile registration.

Validating Profiles Registered in DISABLED State

Validating registered profiles is not required. However, you may validate registered profiles as long as the validation does not prevent the profile from being created.

Registration of DISABLED Profiles that Fail Validation

If the validation of profile in DISABLED state fails, the profile is still registered. Profiles in the DISABLED state may contain errors or the credentials to the target system directory may be unknown, however, this does not prevent the profile from being registered.

Correcting Profile Errors

If you receive errors while registering a profile, for example, due to an incorrect third party directory password, use the `manageSyncProfiles` command line tool to correct the errors in the profile. Refer to "Managing Synchronization Profiles Using manageSyncProfiles" on page 7-16 for more information.

E.3.3 Understanding the diagnostic.log File

This section explains how to understand the Oracle Directory Integration Platform diagnostic.log file, which is located at the following location:

MW_HOME/user_projects/domains/*DOMAIN_NAME*/servers/*NAME_OF_MANAGED_SERVER*/logs/

Note: The file name is *NAME_OF_MANAGED_SERVER*`-diagnostic.log`.

This following is an example diagnostic.log file that is broken into sections and annotated to identify information that will be useful when troubleshooting Oracle Directory Integration Platform. Noteworthy information is shown in **bold type**, and the text **Host: *HOST_NAME: PORT*** indicates the host name and port of the machine on which Oracle Directory Integration Platform is connecting.

Startup Information

The following section of the diagnostic.log file shows information related to Oracle Directory Integration Platform startup. In this section, notice the following:

- **SSL Mode:** indicates the connection mode used for connecting to the Oracle back-end directory. You may see SSL Mode: 1 or SSL Mode: 2. If you see SSL Mode: 2, Oracle Directory Integration Platform uses certificates to connect to the Oracle back-end directory. (Oracle Unified Directory and Oracle Directory Server Enterprise Edition only support SSL mode 2. They do not support mode 1 (non-SSL mode 1).
- **Scheduler initialized** indicates that the profile scheduler has initialized properly. A string indicating that a successful connection to the Oracle back-end directory server follows.
- **Schema objects** are initialized and **profiles** are scheduled for synchronization.

```
[2009-02-18T00:52:27.530-08:00] [wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid:
[ACTIVE].ExecuteThread: '1' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0] Copyright (c) 1982,
2009 Oracle.  All rights reserved

[2009-02-18T00:52:27.550-08:00] [wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid:
[ACTIVE].ExecuteThread: '1' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0] SSL Mode : 1

[2009-02-18T00:52:27.554-08:00] [wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid:
[ACTIVE].ExecuteThread: '1' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId:
```

<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0] **Host: *HOST_NAME:***
PORT

[2009-02-18T00:52:38.104-08:00] [wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid: Scheduler] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0] **Scheduler**
intialized

[2009-02-18T00:52:47.273-08:00] [wls_ods1] [NOTIFICATION] [DIP-10571] [oracle.dip] [tid: Scheduler]
[userId: <anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0] **Connection**
to LDAP Server Successful

[2009-02-18T00:52:47.334-08:00] [wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid: Scheduler] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0] **OBJECT_SCHEMA_**
READER_INITIALIZING

[2009-02-18T00:52:47.508-08:00] [wls_ods1] [NOTIFICATION] [DIP-10572] [oracle.dip] [tid: Scheduler]
[userId: <anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0] **Object**
Schema Reader Initialized.

[2009-02-18T00:52:47.510-08:00] [wls_ods1] [NOTIFICATION] [DIP-10573] [oracle.dip] [tid: Scheduler]
[userId: <anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0] **Event**
Schema Reader Initialized.

[2009-02-18T00:52:48.198-08:00] [wls_ods1] [NOTIFICATION] [DIP-10574] [oracle.dip] [tid: Scheduler]
[userId: <anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0] Data
transfer interface defn initialized

[2009-02-18T00:52:48.213-08:00] [wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid: Scheduler] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0] **INITALIZE_PROVJOBS**

[2009-02-18T00:52:48.773-08:00] [wls_ods1] [NOTIFICATION] [DIP-10566] [oracle.dip] [tid: Scheduler]
[userId: <anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0] [arg:
\n----------EVENT TYPE CONFIGURATION
---------------\n--------------------------------\nEventLDAPChangeType :
ADD,MODIFY,DELETE\nobjectclass:
inetorgperson,orcluserv2\n--------------------------------\nEventLDAPChangeType :
ADD,MODIFY,DELETE\nobjectclass:
orclservicesubscriptiondetail\n--------------------------------\nEventLDAPChangeType :
ADD,MODIFY,DELETE\nobjectclass: *\n--------------------------------\nEventLDAPChangeType :
ADD,MODIFY,DELETE\nobjectclass:
inetorgperson,orcluserv2\n--------------------------------\nEventLDAPChangeType :
ADD,MODIFY,DELETE\nobjectclass:
orclsubscriber\n--------------------------------\nEventLDAPChangeType :
ADD,MODIFY,DELETE\nobjectclass:
orclgroup,orclprivilegegroup,groupofuniquenames,groupofnames\n-----------------------------------
-------------] Print Event Type Configuration...[[
----------EVENT TYPE CONFIGURATION ---------------

EventLDAPChangeType : ADD,MODIFY,DELETE
objectclass: inetorgperson,orcluserv2

EventLDAPChangeType : ADD,MODIFY,DELETE
objectclass: orclservicesubscriptiondetail

EventLDAPChangeType : ADD,MODIFY,DELETE
objectclass: *

EventLDAPChangeType : ADD,MODIFY,DELETE
objectclass: inetorgperson,orcluserv2

```
EventLDAPChangeType : ADD,MODIFY,DELETE
objectclass: orclsubscriber
--------------------------------
EventLDAPChangeType : ADD,MODIFY,DELETE
objectclass: orclgroup,orclprivilegegroup,groupofuniquenames,groupofnames
------------------------------------------------
]]

[2009-02-18T00:52:48.826-08:00] [wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid: Scheduler] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0] INITIALIZE_SYNCJOBS

[2009-02-18T00:52:50.804-08:00] [wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid: Scheduler] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0] Job submission
successfulActiveExport SYNC_JOB 60

[2009-02-18T00:52:50.809-08:00] [wls_ods1] [NOTIFICATION] [EVENT_NOT_ENABLED] [oracle.dip] [tid:
Scheduler] [userId: <anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0]

[2009-02-18T00:52:52.184-08:00] [wls_ods1] [NOTIFICATION] [DIP-10605] [oracle.dip] [tid: Scheduler]
[userId: <anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP#11.1.1.1.0] [arg:
ActiveExport] Profile : ActiveExport added successfully for scheduling.
```

UpdateThread Checking for Changes in Profiles

The following section of the `diagnostic.log` file shows information related to the UpdateThread job, which checks for changes made to synchronization and provisioning profiles. If UpdateThread finds changes, the profile is modified and rescheduled. In this section, notice the following:

```
[2009-02-18T01:20:42.501-08:00] [wls_ods1] [NOTIFICATION] [DIP-10580] [oracle.dip] [tid:
UpdateThread] [userId: <anonymous>] [ecid: 0000Hy8fyF1F0FQ6ubn3EH19ax8V000003,0] [APP:
DIP#11.1.1.1.0] [arg:
(&(objectclass=changelogentry)(changenumber>=3340)(|(targetdn=*cn=Profiles,cn=Provisioning,cn=Direc
tory Integration Platform,cn=Products,cn=OracleContext)(targetdn=*cn=event definitions,cn=directory
integration platform,cn=products,cn=oraclecontext)(targetdn=*cn=object definitions,cn=directory
integration platform,cn=products,cn=oraclecontext)))] Changelog Filter :
(&(objectclass=changelogentry)(changenumber>=3340)(|(targetdn=*cn=Profiles,cn=Provisioning,cn=Direc
tory Integration Platform,cn=Products,cn=OracleContext)(targetdn=*cn=event definitions,cn=directory
integration platform,cn=products,cn=oraclecontext)(targetdn=*cn=object definitions,cn=directory
integration platform,cn=products,cn=oraclecontext)))
```

Profile Initialization

The following section of the `diagnostic.log` file shows information related to profile initialization. In this section, notice that the ActiveImport profile is scheduled:

```
[2009-02-18T02:26:19.604-08:00] [wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid: Scheduler] [userId:
<anonymous>] [ecid: 0000Hy8unSqF0FQ6ubn3EH19ay88000001,0] [APP: dipapp#11.1.1.1.0] INITIALIZE_
SYNCJOBS

[2009-02-18T02:26:19.695-08:00] [wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid: Scheduler] [userId:
<anonymous>] [ecid: 0000Hy8unSqF0FQ6ubn3EH19ay88000001,0] [APP: dipapp#11.1.1.1.0] Job submission
successfulActiveImport SYNC_JOB 60

[2009-02-18T02:26:19.703-08:00] [wls_ods1] [NOTIFICATION] [EVENT_NOT_ENABLED] [oracle.dip] [tid:
Scheduler] [userId: <anonymous>] [ecid: 0000Hy8unSqF0FQ6ubn3EH19ay88000001,0] [APP:
dipapp#11.1.1.1.0]

[2009-02-18T02:26:19.741-08:00] [wls_ods1] [NOTIFICATION] [DIP-10605] [oracle.dip] [tid: Scheduler]
[userId: <anonymous>] [ecid: 0000Hy8unSqF0FQ6ubn3EH19ay88000001,0] [APP: dipapp#11.1.1.1.0] [arg:
```

```
ActiveImport] profile added successfully for scheduling : ActiveImport
```

Database Failure

The following section of the `diagnostic.log` file shows information that appears if the database is not running:

```
Feb 18, 2009 3:01:19 AM org.quartz.impl.jdbcjobstore.JobStoreSupport$ClusterManager manage
SEVERE: ClusterManager: Error managing cluster: Failed to obtain DB connection from data source
'schedulerDS': java.sql.SQLException: Could not retrieve datasource via JNDI url 'jdbc/schedulerDS'
weblogic.jdbc.extensions.PoolDisabledSQLException:
weblogic.common.resourcepool.ResourceDisabledException: Pool schedulerDS is disabled, cannot
allocate resources to applications..
org.quartz.JobPersistenceException: Failed to obtain DB connection from data source 'schedulerDS':
java.sql.SQLException: Could not retrieve datasource via JNDI url 'jdbc/schedulerDS'
weblogic.jdbc.extensions.PoolDisabledSQLException:
weblogic.common.resourcepool.ResourceDisabledException: Pool schedulerDS is disabled, cannot
allocate resources to applications.. [See nested exception: java.sql.SQLException: Could not
retrieve datasource via JNDI url 'jdbc/schedulerDS'
weblogic.jdbc.extensions.PoolDisabledSQLException:
weblogic.common.resourcepool.ResourceDisabledException: Pool schedulerDS is disabled, cannot
allocate resources to applications..]
        at org.quartz.impl.jdbcjobstore.JobStoreSupport.getConnection(JobStoreSupport.java:636)
        at org.quartz.impl.jdbcjobstore.JobStoreTX.getNonManagedTXConnection(JobStoreTX.java:72)
        at org.quartz.impl.jdbcjobstore.JobStoreSupport.doCheckin(JobStoreSupport.java:3070)
        at
org.quartz.impl.jdbcjobstore.JobStoreSupport$ClusterManager.manage(JobStoreSupport.java:3713)
        at
org.quartz.impl.jdbcjobstore.JobStoreSupport$ClusterManager.run(JobStoreSupport.java:3749)
Caused by: java.sql.SQLException: Could not retrieve datasource via JNDI url 'jdbc/schedulerDS'
weblogic.jdbc.extensions.PoolDisabledSQLException:
weblogic.common.resourcepool.ResourceDisabledException: Pool schedulerDS is disabled, cannot
allocate resources to applications..
        at org.quartz.utils.JNDIConnectionProvider.getConnection(JNDIConnectionProvider.java:166)
        at org.quartz.utils.DBConnectionManager.getConnection(DBConnectionManager.java:112)
        at org.quartz.impl.jdbcjobstore.JobStoreSupport.getConnection(JobStoreSupport.java:633)
```

Successful Synchronization OPeration

The following section of the `diagnostic.log` file shows the successful synchronization of a user:

```
QuartzJobListener says: Job ActiveImport Is about to be executed.Wed Feb 18 03:36:00 PST 2009
createChangeRecord:ChangeRecord : ----------
Changetype: ADDRMODIFY
ChangeKey: cn=myuser2,cn=users,dc=imtest,dc=com
Attributes:
Class: null Name: userprincipalname Type: null ChgType: DELETE Value: [ ]
Class: null Name: givenname Type: null ChgType: DELETE Value: [ ]
Class: null Name: employeeid Type: null ChgType: DELETE Value: [ ]
Class: null Name: physicaldeliveryofficename Type: null ChgType: DELETE Value: [ ]
Class: null Name: title Type: null ChgType: DELETE Value: [ ]
Class: null Name: mobile Type: null ChgType: DELETE Value: [ ]
Class: null Name: telephonenumber Type: null ChgType: DELETE Value: [ ]
Class: null Name: facsimiletelephonenumber Type: null ChgType: DELETE Value: [ ]
Class: null Name: l Type: null ChgType: DELETE Value: [ ]
Class: null Name: thumbnailphoto Type: null ChgType: DELETE Value: [ ]
Class: null Name: samaccountname Type: nonbinary ChgType: REPLACE Value: [MyUser2]
Class: null Name: objectsid Type: nonbinary ChgType: REPLACE Value: [[B@1b994c4]
Class: null Name: objectguid Type: nonbinary ChgType: REPLACE Value: [[B@1b990b5]
Class: null Name: distinguishedname Type: nonbinary ChgType: REPLACE Value:
```

```
[CN=MyUser2,CN=Users,DC=imtest,DC=com]
Class: null Name: cn Type: nonbinary ChgType: REPLACE Value: [MyUser2]
Class: null Name: objectclass Type: nonbinary ChgType: REPLACE Value: [top, person,
organizationalPerson, user]
-----------
copying : changeRecord to dstchange for writing
In DIPSYNC: doOneIteration():execMapping status0
QuartzJobListener says: Job ActiveImport was executed.Wed Feb 18 03:36:00 PST 2009
```

E.4 Troubleshooting Integration with Microsoft Active Directory

This section describes how to troubleshoot integration with Microsoft Active Directory. It contains these topics:

- Debugging Windows Native Authentication
- Synchronizing Changes Following a Period when the Oracle Back-end Directory is Unavailable

E.4.1 Debugging Windows Native Authentication

Once you have configured Windows Native Authentication (see "Configuring Windows Native Authentication" on page 18-8), you can enable logging for this feature at run time. Open the `opmn.xml` file, located in *$ORACLE_HOME*`/opmn/conf`, and add the following parameter:

```
-Djazn.debug.log.enable = {true | false}
```

Assigning a value of `true` to the parameter enables debugging while assigning a value of `false` disables it.

The boldface text in the following example show where you should place the parameter in the `opmn.xml` file:

```
<process-type id="OC4J_SECURITY" module-id="OC4J">
  <environment>
    <variable id="DISPLAY" value="sun1.us.oracle.com:0.0"/>
    <variable id="LD_LIBRARY_PATH" value="/private/ora1012/OraHome1/lib"/>
  </environment>
  <module-data>
    <category id="start-parameters">
      <data id="java-options" value="-server -Djazn.debug.log.enable=true
      -Djava.security.policy=/private/ora1012/OraHome1/j2ee/OC4J_SECURITY/
      config/java2.policy -Djava.awt.headless=true -Xmx512m
      -Djava.awt.headless=true"/>
      <data id="oc4j-options" value="-properties"/>
    </category>
    <category id="stop-parameters">
      <data id="java-options" value="-Djava.security.policy=/private/ora1012/
      OraHome1/j2ee/OC4J_SECURITY/config/java2.policy  -Djava.awt.headless=true"/>
    </category>
```

The log is written to the file `OC4J~OC4J_SECURITY~default_island~1`, found at *$ORACLE_HOME*`/opmn/logs`.

Note: When accessing a protected application with Windows Native Authentication, Web browsers automatically return a "401 - Unauthorized" error that is logged by Oracle Enterprise Manager. This is normal behavior and can be safely ignored.

See Also:

- Note: 283268.1—*Troubleshooting Oracle Application Server Single Sign-On Windows Native Authentication* in My Oracle Support (formerly MetaLink) at `http://metalink.oracle.com/`
- The "Problems and Solutions for Windows Native Authentication Errors" section in the Troubleshooting chapter of the *Oracle Enterprise Single Sign-On Suite Plus Administrator's Guide* for more information about Windows Native Authentication errors.

E.4.2 Synchronizing Changes Following a Period when the Oracle Back-end Directory is Unavailable

When the Oracle back-end directory is unavailable, changes are stored in Microsoft Active Directory. The Oracle Password Filter for Microsoft Active Directory attempts to synchronize these entries after connectivity is restored with the Oracle back-end directory. The `SearchDeltaSize` parameter determines how many incremental changes are processed during each iteration in a synchronization cycle. By default, the `SearchDeltaSize` parameter is assigned a value of 500. Depending on how long the Oracle back-end directory is unavailable, the default `SearchDeltaSize` value of 500 may be too low to catch up all of the unsynchronized changes. To resolve this problem, you must create a catchup profile by copying the existing Microsoft Active Directory import synchronization profile and modifying the value assigned to the `SearchDeltaSize` parameter.

To create a catchup synchronization profile:

1. Stop the Oracle Directory Integration Platform.
2. Deactivate the Microsoft Active Directory import synchronization profile using the deactivate operation of the manageSyncProfiles command.
3. Use the `manageSyncProfiles copy` command to create the catchup synchronization profile by copying the import synchronization profile. For example:

   ```
   manageSyncProfiles copy -h myhost.mycompany.com -p 7005 -D weblogic
   -pf existing_import_sync_profile -newpf name_of_new_catchup_sync_profile
   ```

4. Activate the original Microsoft Active Directory import synchronization profile using the activate operation of the manageSyncProfiles command.
5. Start the Oracle Directory Integration Platform.
6. Obtain the current value of the `highestCommittedUSN` by searching the new domain controller's root DSE for the current highest `USNChanged` value (attribute value of the `highestCommittedUSN` attribute of the root DSE):

   ```
   ldapsearch -h host -p port -b "" -s base -D binddn -q \
    DN "objectclass=*" highestCommittedUSN
   ```

Note: You will be prompted for the password.

7. Experiment with the following `ldapsearch` command until you retrieve more than 100 entries but less than 200. Retrieving more than 200 entries may result in an internal buffer overrun.

```
ldapsearch -v -h adhost -p adport -D administrator@domain -q \
-b cn=users,dc=acme,dc=com -s sub \
"(&(objectclass=*)(usnChanged>=delta)(&(usnChanged<=highestCommittedUSN)))" dn
```

Note: You will be prompted for the password.

For example, the following command performs a search using a default search delta size of 500:

```
ldapsearch -v -h adhost -p adport -D administrator@domain -q \
-b cn=users,dc=acme,dc=com -s sub  \
"(&(objectclass=*)(usnChanged>=55010)(&(usnChanged<=55510)))" dn
```

Note: You will be prompted for the password.

8. Create a text file named profile_config.txt that contains the following:

```
[INTERFACEDETAILS]
Package: gsi
Reader: ActiveChgReader
SkipErrorToSyncNextChange: true
SearchDeltaSize: 100000
```

Note: You can also set the `SkipErrorToSyncNextChange` parameter to determine how the Oracle Directory Integration Platform handles an error when processing a change during synchronization. See the "Advanced" section on page 7-5 for more information about the `SkipErrorToSyncNextChange` parameter in synchronization profiles.

9. Use the `update` operation of the `manageSyncProfiles` command to load the profile_config.txt file into the catchup synchronization profile.
10. Use the `activate` operation of the `manageSyncProfiles` command to activate the catchup synchronization profile.

Note: Be sure to continue running the original Microsoft Active Directory import synchronization profile along with the catchup synchronization profile.

11. Allow the catchup synchronization profile to run for at least 12 hours. After all of the backlogged changes are synchronized, use the `deactivate` operation of the `manageSyncProfiles` command to deactivate the catchup synchronization profile.

E.5 Need More Help?

You can find more solutions in My Oracle Support (formerly MetaLink) at http://metalink.oracle.com. If you do not find a solution for your problem, log a service request.

> **See Also:** *Oracle Application Server Release Notes*, available on the Oracle Technology Network: http://www.oracle.com/technology/documentation/index.html

Glossary

access control item (ACI)

An attribute that determines who has what type of access to what directory data. It contains a set of rules for structural access items, which pertain to entries, and content access items, which pertain to attributes. Access to both structural and content access items may be granted to one or more users or groups.

access control list (ACL)

The group of access directives that you define. The directives grant levels of access to specific data for specific clients, or groups of clients, or both.

access control policy point

An entry that contains security directives that apply downward to all entries at lower positions in the **directory information tree (DIT)**.

ACI

See **access control item (ACI)**.

ACL

See **access control list (ACL)**.

ACP

See **access control policy point**.

administrative area

A subtree on a directory server whose entries are under the control (schema, ACL, and collective attributes) of a single administrative authority.

advanced symmetric replication (ASR)

See **Oracle Database Advanced Replication**

agent

An agent transforms data from one of the formats supported by Oracle Directory Integration Platform into a format supported by the connected directory.

anonymous authentication

The process by which the directory authenticates a user without requiring a user name and password combination. Each anonymous user then exercises the privileges specified for anonymous users.

API

See **application program interface (API)**.

application program interface (API)

Programs to access the services of a specified application. For example, LDAP-enabled clients access directory information through programmatic calls available in the LDAP API.

ASR

See **Oracle Database Advanced Replication**.

attribute

An item of information that describes some aspect of an entry. An entry comprises a set of attributes, each of which belongs to an **object class**. Moreover, each attribute has both a *type*, which describes the kind of information in the attribute, and a *value*, which contains the actual data.

attribute configuration file

In an Oracle Directory Integration Platform environment, a file that specifies attributes in a connected directory.

attribute type

The kind of information an attribute contains, for example, `jobTitle`.

attribute uniqueness

An Oracle Internet Directory feature that ensures that no two specified attributes have the same value. It enables applications synchronizing with the enterprise directory to use attributes as unique keys.

attribute value

The particular occurrence of information appearing in that entry. For example, the value for the `jobTitle` attribute could be `manager`.

authentication

The process of verifying the identity of a user, device, or other entity in a computer system, often as a prerequisite to allowing access to resources in a system.

authorization

Permission given to a user, program, or process to access an object or set of objects.

back-end directory

The repository in which Oracle components and third-party applications store and access user identities and credentials. In an Oracle Directory Integration Platform environment, either Oracle Internet Directory, Oracle Unified Directory, or Oracle Directory Server Enterprise Edition can be utilized as the Oracle back-end directory.

binding

The process of authenticating to a directory.

bootstrapping

The initial migration of data between a connected directory and the Oracle back-end directory. Performing directory bootstrapping when you first deploy Oracle Directory

Integration Platform can save time if you need to move large amounts of directory data.

central directory

In an Oracle Directory Integration Platform environment, the directory that acts as the central repository.

certificate

An ITU x.509 v3 standard data structure that securely binds an identity to a public key. A certificate is created when an entity's public key is signed by a trusted identity: a **certificate authority (CA)**. This certificate ensures that the entity's information is correct and that the public key actually belongs to that entity.

certificate authority (CA)

A trusted third party that certifies that other entities—users, databases, administrators, clients, servers—are who they say they are. The certificate authority verifies the user's identity and grants a certificate, signing it with the certificate authority's private key.

certificate chain

An ordered list of certificates containing an end-user or subscriber certificate and its certificate authority certificates.

change logs

A database that records changes made to a directory server.

cipher suite

In SSL, a set of authentication, encryption, and data integrity algorithms used for exchanging messages between network nodes. During an SSL handshake, the two nodes negotiate to see which cipher suite they will use when transmitting messages back and forth.

cluster

A collection of interconnected computers that are used as a single computing resource. Hardware clusters provide high availability and scalability.

cold backup

The procedure to add a new **DSA** node to an existing replicating system by using the database copy procedure.

concurrency

The ability to handle multiple requests simultaneously. Threads and processes are examples of concurrency mechanisms.

concurrent clients

The total number of clients that have established a session with the Oracle back-end directory.

concurrent operations

The number of operations that are being run on the directory from all of the concurrent clients. Note that this is not necessarily the same as the concurrent clients, because some of the clients may be keeping their sessions idle.

connect descriptor

A specially formatted description of the destination for a network connection. A connect descriptor contains destination service and network route information.

The destination service is indicated by using its service name for the Oracle Database or its Oracle System Identifier (SID) for Oracle release 8.0 or version 7 databases. The network route provides, at a minimum, the location of the listener through use of a network address.

connected directory

In an Oracle Directory Integration Platform environment, an information repository requiring full synchronization of data between the Oracle back-end directory and itself—for example, an Oracle human Resources database.

connector

A connectivity solution that Oracle Directory Integration Platform uses for synchronization between the Oracle back-end directory and a connected directory. At a minimum, a connector consists of a directory integration profile containing all the configuration information required for synchronization.

consumer

A directory server that is the destination of replication updates. Sometimes called a slave.

contention

Competition for resources.

context prefix

The **DN** of the root of a **naming context**.

cryptography

The practice of encoding and decoding data, resulting in secure messages.

Data Encryption Standard (DES)

A block cipher developed by IBM and the U.S. government in the 1970's as an official standard.

data integrity

The guarantee that the contents of the message received were not altered from the contents of the original message sent.

decryption

The process of converting the contents of an encrypted message (cipher text) back into its original readable format (plain text).

default knowledge reference

A **knowledge reference** that is returned when the base object is not in the directory, and the operation is performed in a naming context not held locally by the server. A default knowledge reference typically sends the user to a server that has more knowledge about the directory partitioning arrangement.

default identity management realm

In a hosted environment, one enterprise—for example, an application service provider—makes Oracle components available to multiple enterprises and stores

information for them. In such hosted environments, the enterprise performing the hosting is called the default identity management realm, and the enterprises that are hosted are each associated with their own identity management realm in the DIT.

default realm location

An attribute in the root Oracle Context that identifies the root of the default identity management realm.

delegated administrator

In a hosted environment, one enterprise—for example, an application service provider—makes Oracle components available to multiple other enterprises and stores information for them. In such an environment, a global administrator performs activities that span the entire directory. Other administrators—called delegated administrators—may exercise roles in specific identity management realms, or for specific applications.

DES

See **Data Encryption Standard (DES)**.

DIB

See **directory information base (DIB)**.

directory information base (DIB)

The complete set of all information held in the directory. The DIB consists of entries that are related to each other hierarchically in a **directory information tree (DIT)**.

directory information tree (DIT)

A hierarchical, tree-like structure consisting of the DNs of the entries.

directory integration profile

In an Oracle Directory Integration Platform environment, an entry in the Oracle back-end directory that describes how Oracle Directory Integration Platform communicates with external systems and what is communicated.

directory integration server

In an Oracle Directory Integration Platform environment, the server that drives the synchronization of data between the Oracle back-end directory and a **connected directory**.

directory naming context

See **naming context**.

directory provisioning profile

A special kind of **directory integration profile** that describes the nature of provisioning-related notifications that Oracle Directory Integration Platform sends to the directory-enabled applications.

directory replication group (DRG)

The directory servers participating in a replication agreement.

directory server instance

A discrete invocation of a directory server. Different invocations of a directory server, each started with the same or different configuration set entries and startup flags, are said to be different directory server instances.

directory-specific entry (DSE)

An entry specific to a directory server. Different directory servers may hold the same DIT name, but have different contents—that is, the contents can be specific to the directory holding it. A DSE is an entry with contents specific to the directory server holding it.

directory synchronization profile

A special kind of **directory integration profile** that describes how synchronization is carried out between the Oracle back-end directory and an external system.

directory system agent (DSA)

The X.500 term for a directory server.

distinguished name (DN)

The unique name of a directory entry. It comprises all of the individual names of the parent entries back to the root.

DIS

See **directory integration server**.

DIT

See **directory information tree (DIT)**.

DN

See **distinguished name (DN)**.

DRG

See **directory replication group (DRG)**.

DSA

See **directory system agent (DSA)**.

DSE

See **directory-specific entry (DSE)**.

DSA-specific entries. Different DSAs may hold the same DIT name, but have different contents. That is, the contents can be specific to the DSA holding it. A DSE is an entry with contents specific to the DSA holding it.

encryption

The process of disguising the contents of a message and rendering it unreadable (ciphertext) to anyone except for the intended recipient.

entry

The building block of a directory, it contains information about an object of interest to directory users.

export agent

In an Oracle Directory Integration Platform environment, an agent that exports data out of the Oracle back-end directory.

export data file

In an Oracle Directory Integration Platform environment, the file that contains data exported by an **export agent**.

export file

See **export data file**.

external agent

A directory integration agent that is independent of Oracle Directory Integration Platform. Oracle Directory Integration Platform does not provide scheduling, mapping, or error handling services for it. An external agent is typically used when a third party metadirectory solution is integrated with the Oracle Directory Integration Platform.

failover

The process of failure recognition and recovery. In an Oracle Application Server Cold Failover Cluster (Infrastructure), an application running on one cluster node is transparently migrated to another cluster node. During this migration, clients accessing the service on the cluster see a momentary outage and may need to reconnect once the failover is complete.

fan-out replication

Also called a point-to-point replication. A type of replication in which a supplier replicates directly to a consumer. That consumer can then replicate to one or more other consumers. The replication can be either full or partial.

filter

A method of qualifying data, usually data that you are seeking. Filters are always expressed as DNs, for example: `cn=susie smith,o=acme,c=us`.

global administrator

In a hosted environment, one enterprise—for example, an application service provider—makes Oracle components available to multiple other enterprises and stores information for them. In such an environment, a global administrator performs activities that span the entire directory.

global unique identifier (GUID)

An identifier generated by the system and inserted into an entry when the entry is added to the directory. In a multimaster replicated environment, the GUID, not the DN, uniquely identifies an entry. The GUID of an entry cannot be modified by a user.

grace login

A login occurring within the specified period before password expiration.

group search base

In the Oracle back-end directory default DIT, the node in the identity management realm under which all the groups can be found.

guest user

One who is not an anonymous user, and, at the same time, does not have a specific user entry.

GUID

See **global unique identifier (GUID)**.

handshake

A protocol two computers use to initiate a communication session.

hash

A number generated from a string of text with an algorithm. The hash value is substantially smaller than the text itself. Hash numbers are used for security and for faster access to data.

identity management

The process by which the complete security life cycle for network entities is managed in an organization. It typically refers to the management of an organization's application users, where steps in the security life cycle include account creation, suspension, privilege modification, and account deletion. The network entities managed can also include devices, processes, applications, or anything else that needs to interact in a networked environment. Entities managed by an identity management process can also include users outside of the organization, for example customers, trading partners, or Web services.

identity management realm

A collection of identities, all of which are governed by the same administrative policies. In an enterprise, all employees having access to the intranet may belong to one realm, while all external users who access the public applications of the enterprise may belong to another realm. An identity management realm is represented in the directory by a specific entry with a special object class associated with it.

identity management realm-specific Oracle Context

An Oracle Context contained in each identity management realm. It stores the following information:

- User naming policy of the identity management realm—that is, how users are named and located
- Mandatory authentication attributes
- Location of groups in the identity management realm
- Privilege assignments for the identity management realm—for example: who has privileges to add more users to the Realm.
- Application specific data for that Realm including authorizations

import agent

In an Oracle Directory Integration Platform environment, an agent that imports data into the Oracle back-end directory.

import data file

In an Oracle Directory Integration Platform environment, the file containing the data imported by an **import agent**.

inherit

When an object class has been derived from another class, it also derives, or inherits, many of the characteristics of that other class. Similarly, an attribute subtype inherits the characteristics of its supertype.

instance

See **directory server instance**.

integrity

The guarantee that the contents of the message received were not altered from the contents of the original message sent.

Internet Engineering Task Force (IETF)

The principal body engaged in the development of new Internet standard specifications. It is an international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet.

Internet Message Access Protocol (IMAP)

A protocol allowing a client to access and manipulate electronic mail messages on a server. It permits manipulation of remote message folders, also called mailboxes, in a way that is functionally equivalent to local mailboxes.

key

A string of bits used widely in cryptography, allowing people to encrypt and decrypt data; a key can be used to perform other mathematical operations as well. Given a cipher, a key determines the mapping of the plaintext to the ciphertext.

key pair

A **public key** and its associated **private key**.

See **public/private key pair**.

knowledge reference

The access information (name and address) for a remote **DSA** and the name of the **DIT** subtree that the remote DSA holds. Knowledge references are also called referrals.

latency

The time a client has to wait for a given directory operation to complete. Latency can be defined as wasted time. In networking discussions, latency is defined as the travel time of a packet from source to destination.

LDAP

See **Lightweight Directory Access Protocol (LDAP)**.

LDIF

See **LDAP Data Interchange Format (LDIF)**.

Lightweight Directory Access Protocol (LDAP)

A standard, extensible directory access protocol. It is a common language that LDAP clients and servers use to communicate. The framework of design conventions supporting industry-standard directory products, such as the Oracle Internet Directory.

LDAP Data Interchange Format (LDIF)

The set of standards for formatting an input file for any of the LDAP command-line utilities.

logical host

In an Oracle Application Server Cold Failover Cluster (Infrastructure), one or more disk groups and pairs of host names and IP addresses. It is mapped to a physical host in the cluster. This physical host impersonates the host name and IP address of the logical host

man-in-the-middle

A security attack characterized by the third-party, surreptitious interception of a message. The third-party, the *man-in-the-middle*, decrypts the message, re-encrypts it (with or without alteration of the original message), and retransmits it to the originally-intended recipient—all without the knowledge of the legitimate sender and receiver. This type of security attack works only in the absence of **authentication**.

mapping rules file

In an Oracle Directory Integration Platform environment, the file that specifies mappings between the Oracle back-end directory attributes and those in a **connected directory**.

master definition site (MDS)

In replication, a master definition site is the Oracle Internet Directory database from which the administrator runs the configuration scripts.

master site

In replication, a master site is any site other than the master definition site that participates in LDAP replication.

matching rule

In a search or compare operation, determines equality between the attribute value sought and the attribute value stored. For example, matching rules associated with the `telephoneNumber` attribute could cause "(650) 123-4567" to be matched with either "(650) 123-4567" or "6501234567" or both. When you create an attribute, you associate a matching rule with it.

MD4

A one-way hash function that produces a 128-bit hash, or message digest. If as little as a single bit value in the file is modified, the MD4 checksum for the file will change. Forgery of a file in a way that will cause MD4 to generate the same result as that for the original file is considered extremely difficult.

MD5

An improved version of MD4.

MDS

See **master definition site (MDS)**

metadirectory

A directory solution that shares information between all enterprise directories, integrating them into one virtual directory. It centralizes administration, thereby

reducing administrative costs. It synchronizes data among directories, thereby ensuring that it is consistent and up-to-date across the enterprise.

MTS

See **shared server**

multimaster replication

Also called peer-to-peer or *n*-way replication, a type of replication that enables multiple sites, acting as equals, to manage groups of replicated data. In a multimaster replication environment, each node is both a supplier and a consumer node, and the entire directory is replicated on each node.

naming attribute

The attribute used to compose the RDN of a new user entry created through Oracle Delegated Administration Services or Oracle Internet Directory Java APIs. The default value for this is `cn`.

naming context

A subtree that resides entirely on one server. It must be contiguous, that is, it must begin at an entry that serves as the top of the subtree, and extend downward to either leaf entries or **knowledge reference**s (also called referrals) to subordinate naming contexts. It can range in size from a single entry to the entire DIT.

native agent

In an Oracle Directory Integration Platform environment, an agent that runs under the control of the **directory integration server**. It is in contrast to an **external agent**.

net service name

A simple name for a service that resolves to a connect descriptor. Users initiate a connect request by passing a user name and password along with a net service name in a connect string for the service to which they wish to connect:

```
CONNECT username/password@net_service_name
```

Depending on your needs, net service names can be stored in a variety of places, including:

- Local configuration file, `tnsnames.ora`, on each client
- Directory server
- Oracle Names server
- External naming service, such as NDS, NIS or CDS

nickname attribute

The attribute used to uniquely identify a user in the entire directory. The default value for this is `uid`. Applications use this to resolve a simple user name to the complete distinguished name. The user nickname attribute cannot have multiple values—that is, a given user cannot have multiple nicknames stored under the same attribute name.

object class

A named group of attributes. When you want to assign attributes to an entry, you do so by assigning to that entry the object classes that hold those attributes.

All objects associated with the same object class share the same attributes.

OEM

See **Oracle Enterprise Manager Fusion Middleware Control**.

OID Control Utility

A command-line tool for issuing run-server and stop-server commands. The commands are interpreted and executed by the **OID Monitor** process.

OID Database Password Utility

The utility used to change the password with which Oracle Internet Directory connects to an Oracle database.

OID Monitor

The Oracle Internet Directory component that initiates, monitors, and terminates the Oracle directory server processes. It also controls the replication server if one is installed, and Oracle Directory Integration Platform.

one-way function

A function that is easy to compute in one direction but quite difficult to reverse compute, that is, to compute in the opposite direction.

one-way hash function

A **one-way function** that takes a variable sized input and creates a fixed size output.

Oracle Call Interface (OCI)

An application programming interface (API) that enables you to create applications that use the native procedures or function calls of a third-generation language to access an Oracle database server and control all phases of SQL statement execution.

Oracle Delegated Administration Services

A set of individual, predefined services—called Oracle Delegated Administration Services units—for performing directory operations on behalf of a user. Oracle Internet Directory Self-Service Console makes it easier to develop and deploy administration solutions for both Oracle and third-party applications that use Oracle Internet Directory.

Oracle Directory Integration Platform

A component of **Oracle Internet Directory**. It is a framework developed to integrate applications around a central LDAP directory like Oracle Internet Directory.

Oracle Enterprise Manager Fusion Middleware Control

A separate Oracle product that combines a graphical console, agents, common services, and tools to provide an integrated and comprehensive systems management platform for managing Oracle products.

Oracle Identity Management

An infrastructure enabling deployments to manage centrally and securely all enterprise identities and their access to various applications in the enterprise.

Oracle Internet Directory

A general purpose directory service that enables retrieval of information about dispersed users and network resources. It combines Lightweight Directory Access Protocol (LDAP) Version 3 with the high performance, scalability, robustness, and availability of Oracle Database.

Oracle Net Services

The foundation of the Oracle family of networking products, allowing services and their client applications to reside on different computers and communicate. The main function of Oracle Net Services is to establish network sessions and transfer data between a client application and a server. Oracle Net Services is located on each computer in the network. Once a network session is established, Oracle Net Services acts as a data courier for the client and the server.

Oracle PKI certificate usages

Defines Oracle application types that a **certificate** supports.

Oracle Wallet Manager

A Java-based application that security administrators use to manage public-key security credentials on clients and servers.

Oracle Database Advanced Replication

A feature in the Oracle Database that enables database tables to be kept synchronized across two Oracle databases.

other information repository

In an Oracle Directory Integration Platform environment, in which Oracle Internet Directory serves as the **central directory**, any information repository except Oracle Internet Directory.

partition

A unique, nonoverlapping directory naming context that is stored on one directory server.

peer-to-peer replication

Also called multimaster replication or *n*-way replication. A type of replication that enables multiple sites, acting as equals, to manage groups of replicated data. In such a replication environment, each node is both a supplier and a consumer node, and the entire directory is replicated on each node.

PKCS #12

A **public-key encryption** standard (PKCS). RSA Data Security, Inc. PKCS #12 is an industry standard for storing and transferring personal authentication credentials—typically in a format called a **wallet**.

plaintext

Message text that has not been encrypted.

point-to-point replication

Also called fan-out replication. A type of replication in which a supplier replicates directly to a consumer. That consumer can then replicate to one or more other consumers. The replication can be either full or partial.

primary node

In an Oracle Application Server Cold Failover Cluster (Infrastructure), the cluster node on which the application runs at any given time.

private key

In public-key cryptography, this key is the secret key. It is primarily used for decryption, and it is also used for encryption with digital signatures.

provisioning agent

An application or process that translates Oracle-specific provisioning events to external or third-party application-specific events.

provisioned applications

Applications in an environment where user and group information is centralized in Oracle Internet Directory. These applications are typically interested in changes to that information in Oracle Internet Directory.

profile

See **directory integration profile**.

proxy user

A kind of user typically employed in an environment with a middle tier, such as a firewall. In this environment, the end user authenticates to the middle tier. The middle tier then logs into the directory on the end user's behalf. A proxy user has the privilege to switch identities and, once it has logged in to the directory, switches to the end user's identity. It then performs operations on the end user's behalf, using the authorization appropriate to that particular end user.

public key

In public-key cryptography, this key is made public to all; it is primarily used for encryption, but it can be used for verifying signatures.

public-key cryptography

Cryptography based on methods involving a public key and a private key.

public-key encryption

The process in which the sender of a message encrypts the message with the public key of the recipient. Upon delivery, the message is decrypted by the recipient using the recipient's private key.

public/private key pair

A mathematically related set of two numbers where one is called the private key and the other is called the public key. Public keys are typically made widely available, while private keys are available only to their owners. Data encrypted with a public key can only be decrypted with its associated private key and vice versa. Data encrypted with a public key cannot be decrypted with the same public key.

realm search base

An attribute in the root Oracle Context that identifies the entry in the DIT that contains all identity management realms. This attribute is used when mapping a simple realm name to the corresponding entry in the directory.

referral

Information that a directory server provides to a client and which points to other servers the client must contact to find the information it is requesting.

See also **knowledge reference**.

relational database

A structured collection of data that stores data in tables consisting of one or more rows, each containing the same set of columns. Oracle makes it very easy to link the data in multiple tables. This is what makes Oracle a relational database management system, or RDBMS. It stores data in two or more tables, and enables you to define relationships among the tables. The link is based on one or more fields common to both tables.

replica

Each copy of a naming context that is contained within a single server.

RDN

See **relative distinguished name (RDN)**.

registry entry

An entry containing runtime information associated with invocations of Oracle directory servers, called a **directory server instance**. Registry entries are stored in the directory itself, and remain there until the corresponding directory server instance stops.

relative distinguished name (RDN)

The local, most granular-level entry name. It has no other qualifying entry names that would serve to uniquely address the entry. In the example, `cn=Smith,o=acme,c=US`, the RDN is `cn=Smith`.

remote master site (RMS)

In a replicated environment, any site, other than the **master definition site (MDS)**, that participates in Oracle Database Advanced Replication.

replication agreement

A special directory entry that represents the replication relationship among the directory servers in a **directory replication group (DRG)**.

response time

The time between the submission of a request and the completion of the response.

root DSE

See **root directory specific entry**.

root directory specific entry

An entry storing operational information about the directory. The information is stored in a number of attributes.

Root Oracle Context

In the Oracle Identity Management infrastructure, the Root Oracle Context is an entry in Oracle Internet Directory containing a pointer to the default identity management realm in the infrastructure. It also contains information on how to locate an identity management realm given a simple name of the realm.

SASL

See **Simple Authentication and Security Layer (SASL)**.

scalability

The ability of a system to provide throughput in proportion to, and limited only by, available hardware resources.

schema

The collection of attributes, object classes, and their corresponding matching rules.

secondary node

In an Oracle Application Server Cold Failover Cluster (Infrastructure), the cluster node to which an application is moved during a failover.

Secure Hash Algorithm (SHA)

An algorithm that takes a message of less than 264 bits in length and produces a 160-bit message digest. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks.

Secure Socket Layer (SSL)

An industry standard protocol designed by Netscape Communications Corporation for securing network connections. SSL provides authentication, encryption, and data integrity using public key infrastructure (PKI).

service time

The time between the initiation of a request and the completion of the response to the request.

session key

A key for symmetric-key cryptosystems that is used for the duration of one message or communication session.

SGA

See **System Global Area (SGA)**.

SHA

See **Secure Hash Algorithm (SHA)**.

shared server

A server that is configured to allow many user processes to share very few server processes, so the number of users that can be supported is increased. With shared server configuration, many user processes connect to a dispatcher. The dispatcher directs multiple incoming network session requests to a common queue. An idle shared server process from a shared pool of server processes picks up a request from the queue. This means a small pool of server processes can server a large amount of clients. Contrast with dedicated server.

sibling

An entry that has the same parent as one or more other entries.

simple authentication

The process by which the client identifies itself to the server by means of a DN and a password which are not encrypted when sent over the network. In the simple authentication option, the server verifies that the DN and password sent by the client match the DN and password stored in the directory.

Simple Authentication and Security Layer (SASL)

A method for adding authentication support to connection-based protocols. To use this specification, a protocol includes a command for identifying and authenticating a user to a server and for optionally negotiating a security layer for subsequent protocol interactions. The command has a required argument identifying a SASL mechanism.

single key-pair wallet

A **PKCS #12**-format **wallet** that contains a single user **certificate** and its associated **private key**. The **public key** is imbedded in the certificate.

slave

See **consumer**.

SLAPD

Standalone LDAP daemon.

smart knowledge reference

A **knowledge reference** that is returned when the knowledge reference entry is in the scope of the search. It points the user to the server that stores the requested information.

specific administrative area

Administrative areas control:

- Subschema administration
- Access control administration
- Collective attribute administration

A *specific* administrative area controls one of these aspects of administration. A specific administrative area is part of an autonomous administrative area.

sponsor node

In replication, the node that is used to provide initial data to a new node.

SSL

See **Secure Socket Layer (SSL)**.

subACLSubentry

A specific type of subentry that contains ACL information.

subclass

An object class derived from another object class. The object class from which it is derived is called its **superclass**.

subentry

A type of entry containing information applicable to a group of entries in a subtree. The information can be of these types:

- Access control policy points
- Schema rules
- Collective attributes

Subentries are located immediately below the root of an administrative area.

subordinate reference

A knowledge reference pointing downward in the DIT to a naming context that starts immediately below an entry.

subschema DN

The list of DIT areas having independent schema definitions.

subSchemaSubentry

A specific type of **subentry** containing schema information.

subtype

An attribute with one or more options, in contrast to that same attribute without the options. For example, a `commonName` (`cn`) attribute with American English as an option is a subtype of the `commonName` (`cn`) attribute without that option. Conversely, the `commonName` (`cn`) attribute without an option is the **supertype** of the same attribute with an option.

super user

A special directory administrator who typically has full access to directory information.

superclass

The object class from which another object class is derived. For example, the object class `person` is the superclass of the object class `organizationalPerson`. The latter, namely, `organizationalPerson`, is a **subclass** of `person` and inherits the attributes contained in `person`.

superior reference

A knowledge reference pointing upward to a DSA that holds a naming context higher in the DIT than all the naming contexts held by the referencing DSA.

supertype

An attribute without options, in contrast to the same attribute with one or more options. For example, the `commonName` (`cn`) attribute without an option is the supertype of the same attribute with an option. Conversely, a `commonName` (`cn`) attribute with American English as an option is a **subtype** of the `commonName` (`cn`) attribute without that option.

supplier

In replication, the server that holds the master copy of the naming context. It supplies updates from the master copy to the **consumer** server.

System Global Area (SGA)

A group of shared memory structures that contains data and control information for one Oracle Database instance. If multiple users are concurrently connected to the same instance, the data in the instance SGA is shared among the users. Consequently, the SGA is sometimes referred to as the shared global area. The combination of the background processes and memory buffers is called an Oracle instance.

system operational attribute

An attribute holding information that pertains to the operation of the directory itself. Some operational information is specified by the directory to control the server, for example, the timestamp for an entry. Other operational information, such as access

information, is defined by administrators and is used by the directory program in its processing.

TLS

See **Transport Layer Security (TLS)**.

think time

The time the user is not engaged in actual use of the processor.

throughput

The number of requests processed by the Oracle back-end directory for each unit of time. This is typically represented as operations per second.

Transport Layer Security (TLS)

A protocol providing communications privacy over the Internet. The protocol enables client/server applications to communicate in a way that prevents eavesdropping, tampering, or message forgery.

trusted certificate

A third-party identity that is qualified with a level of trust. The trust is used when an identity is being validated as the entity it claims to be. Typically, the certificate authorities you trust issue user certificates.

trustpoint

See **trusted certificate**.

UTF-16

The 16-bit encoding of **Unicode**.The Latin-1 characters are the first 256 code points in this standard.

Unicode

A type of universal character set, a collection of 64K characters encoded in a 16-bit space. It encodes nearly every character in most existing character set standard, covering most written scripts used in the world. It is owned and defined by Unicode Inc. Unicode is canonical encoding which means its value can be passed to different locales. It does not guarantee a round-trip conversion between it and every Oracle character set without information loss.

UNIX Crypt

The UNIX encryption algorithm.

user search base

In the Oracle Internet Directory default DIT, the node in the identity management realm under which all the users are placed.

UTC (Coordinated Universal Time)

The standard time common to every place in the world. Formerly, and widely called Greenwich Mean Time (GMT) and World Time, UTC nominally reflects the mean solar time along the Earth's prime meridian. UTC is indicated by a *z* at the end of the value, for example, 200011281010z.

UTF-8

A variable-width, 8-bit encoding of **Unicode** that uses sequences of 1, 2, 3, or 4 bytes for each character. Characters from 0-127 (the 7-bit ASCII characters) are encoded with one byte, characters from 128-2047 require two bytes, characters from 2048-65535 require three bytes, and characters beyond 65535 require four bytes. The Oracle character set name for this is AL32UTF8 (for the Unicode 3.1 standard).

virtual host name

In an Oracle Application Server Cold Failover Cluster (Infrastructure), the host name corresponding to this virtual IP address.

virtual IP address

In an Oracle Application Server Cold Failover Cluster (Infrastructure), each physical node has its own physical IP address and physical host name. To present a single system image to the outside world, the cluster uses a dynamic IP address that can be moved to any physical node in the cluster. This is called the virtual IP address.

wallet

An abstraction used to store and manage security credentials for an individual entity. It implements the storage and retrieval of credentials for use with various cryptographic services. A wallet resource locator (WRL) provides all the necessary information to locate the wallet.

wait time

The time between the submission of the request and initiation of the response.

X.509

A popular format from ISO used to sign public keys.