

# Oracle® Fusion Middleware

## Administering Oracle Directory Integration Platform



12c (12.2.1.3.0)

E81311-08

July 2020

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

ORACLE®

Oracle Fusion Middleware Administering Oracle Directory Integration Platform, 12c (12.2.1.3.0)

E81311-08

Copyright © 2018, 2020, Oracle and/or its affiliates.

Primary Author: Showvik Roychowdhuri

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

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

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

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

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

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

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

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

# Contents

## Preface

---

Audience	xxv
Documentation Accessibility	xxv
Related Documentation	xxv
Conventions	xxvi

## What's New in Oracle Directory Integration Platform?

---

New and Changed Features for 12c (12.2.1.3.0)	xxviii
New and Deprecated Terminology for Oracle Directory Integration Platform 12c	xxviii

## Part I Getting Started with Oracle Directory Integration Platform

---

### 1 Introduction to Oracle Directory Integration Platform

---

1.1	Why Oracle Directory Integration Platform?	1-1
1.2	Oracle Directory Integration Platform Installation Options	1-3
1.3	Understanding the Differences Between Synchronization and Provisioning	1-3
1.3.1	About Synchronization	1-4
1.3.2	About Provisioning	1-5
1.3.3	How Synchronization and Provisioning Differ	1-5
1.4	Understanding Components Involved in Oracle Directory Integration Platform Integration	1-6
1.4.1	About Oracle Back-End Directory	1-6
1.4.2	About Oracle Directory Integration Platform	1-7
1.4.2.1	Understanding the Oracle Directory Integration Platform Server	1-8
1.4.2.2	Understanding the Oracle Directory Integration Platform Synchronization Service	1-8
1.4.2.3	Understanding the Oracle Directory Integration Platform Provisioning Service	1-10

## 2 Security Features in Oracle Directory Integration Platform

---

2.1	Overview of Authentication in Oracle Directory Integration Platform	2-1
2.1.1	Secure Sockets Layer and Oracle Directory Integration Platform	2-1
2.1.2	Oracle Directory Integration Platform Authentication in SSL Mode	2-2
2.1.3	Transport Layer Security Protocol and Cipher Suites	2-3
2.1.3.1	Supported Out-of-Box Cipher Suites	2-3
2.1.3.2	Disabled Cipher Suites in Oracle Directory Integration Platform	2-5
2.1.3.3	Adding Cipher Suites into Oracle Directory Integration Platform	2-5
2.1.3.4	Specify the SSL/TLS Protocol Version	2-7
2.1.4	Profile Authentication	2-8
2.2	About Access Control and Authorization and Oracle Directory Integration Platform	2-9
2.2.1	Understanding Access Controls for the Oracle Directory Integration Platform	2-9
2.2.2	Understanding Access Controls for Profiles	2-10
2.3	About Data Integrity and Oracle Directory Integration Platform	2-10
2.4	About Data Privacy and Oracle Directory Integration Platform	2-11
2.5	About Tools Security and Oracle Directory Integration Platform	2-11
2.6	Credential Store Framework for Oracle Directory Integration Platform	2-11

## Part II General Administration of Oracle Directory Integration Platform

---

### 3 Administering Oracle Directory Integration Platform

---

3.1	Graphical Tools for Administering Oracle Directory Integration Platform	3-1
3.1.1	Using Fusion Middleware Control	3-1
3.1.1.1	About Oracle Directory Integration Platform Home Page	3-2
3.1.2	Using Oracle Unified Directory Services Manager for Oracle Unified Directory	3-2
3.1.3	Using Oracle Directory Services Manager for Oracle Internet Directory	3-2
3.1.4	Using Directory Service Control Center for Oracle Directory Server Enterprise Edition	3-3
3.2	Command-Line Tools for Administering Oracle Directory Integration Platform	3-3

### 4 Managing the Oracle Directory Integration Platform

---

4.1	Understanding Operational Information About the Oracle Directory Integration Platform	4-2
4.1.1	What are Directory Integration Profiles?	4-2
4.1.2	Understanding Oracle Directory Integration Platform Event Propagation in a Multimaster Oracle Back-end Directory Replication Environment	4-2

4.1.2.1	Synchronizing Directory in an Oracle Back-end Directory Multimaster Replication Environment	4-3
4.1.2.2	About Directory Provisioning in an Oracle Unified Directory or Oracle Internet Directory Multimaster Replication Environment	4-3
4.2	Viewing Oracle Directory Integration Platform Status and Registration Information	4-4
4.2.1	Viewing the Status of Oracle Directory Integration Platform Using the dipStatus Utility	4-4
4.2.2	Viewing Oracle Directory Integration Platform Registration Information Using the ldapsearch Utility	4-4
4.3	Managing Oracle Directory Integration Platform Using Fusion Middleware Control	4-5
4.3.1	Viewing Oracle Directory Integration Platform Runtime Information Using Fusion Middleware Control	4-5
4.3.2	Starting Oracle Directory Integration Platform with Fusion Middleware Control	4-6
4.3.3	Stopping Oracle Directory Integration Platform with Fusion Middleware Control	4-6
4.3.4	Managing the Oracle Directory Integration Platform Server Configuration	4-7
4.3.5	Managing Oracle Directory Integration Platform Logging Using Fusion Middleware Control	4-8
4.3.6	Auditing Oracle Directory Integration Platform Using Fusion Middleware Control	4-9
4.4	Starting and Stopping Oracle Directory Integration Platform Using WLST	4-9
4.5	Manage Oracle Directory Integration Platform Using manageDIPServerConfig	4-9
4.6	About Oracle Unified Directory Configuration for SSL Mode	4-10
4.7	About Oracle Directory Server Enterprise Edition Configuration for SSL Mode	4-10
4.8	About Oracle Internet Directory Configuration for SSL Mode	4-10
4.9	Managing the SSL Certificates of Back-End Directories and Connected Directories	4-11
4.10	About Oracle Directory Integration Platform in a High Availability Scenario	4-12
4.11	Understanding How to Manage Oracle Directory Integration Platform in a Replicated Environment	4-12
4.12	dipStatus Utility	4-12
4.13	manageDIPServerConfig Utility	4-13

## Part III Configuring Oracle Back-End Directory

---

### 5 Configuring Oracle Unified Directory

---

5.1	Before You Configure Oracle Unified Directory as the Back-End Directory	5-1
5.2	Configuring Oracle Unified Directory (Non-SSL) for Oracle Directory Integration Platform	5-1
5.2.1	Installing Oracle Unified Directory	5-2

5.2.2	Configuring Oracle Unified Directory	5-2
5.2.3	Creating Oracle Unified Directory Suffixes	5-2
5.2.4	Enabling External Change Log	5-3
5.2.5	Configuring the Oracle WebLogic Server Domain for Oracle Directory Integration Platform with Oracle Unified Directory	5-4
5.2.5.1	Configuring Oracle Directory Integration Platform with Oracle Unified Directory in an Existing WebLogic Domain	5-4
5.2.5.2	Configuring Oracle Directory Integration Platform and Oracle Unified Directory in a New Oracle WebLogic Server Domain	5-5
5.2.6	Starting the Servers	5-8
5.2.7	Configuring Oracle Directory Integration Platform for Oracle Unified Directory	5-8
5.2.8	Adding Access Control Instructions (ACIs) for Oracle Unified Directory	5-11
5.3	Configuring Oracle Unified Directory (SSL) for Oracle Directory Integration Platform	5-11
5.3.1	Configuring Oracle Unified Directory for SSL	5-12
5.3.2	Configuring Oracle Directory Integration Platform for Oracle Unified Directory SSL	5-12
5.4	Verifying Oracle Directory Integration Platform	5-13

## 6 Configuring Oracle Internet Directory

---

6.1	Before You Configure Oracle Internet Directory as the Back-End Directory	6-1
6.2	Configuring the Oracle WebLogic Server Domain for Oracle Directory Integration Platform with Oracle Internet Directory	6-1
6.2.1	Configuring Oracle Directory Integration Platform with Oracle Internet Directory in an Existing WebLogic Domain	6-2
6.2.2	Configuring Oracle Directory Integration Platform and Oracle Internet Directory in a New Oracle WebLogic Server Domain	6-4
6.3	Configuring Oracle Internet Directory (SSL) for Oracle Directory Integration Platform	6-6
6.3.1	Configuring Oracle Internet Directory for SSL	6-6
6.3.2	Configuring Oracle Directory Integration Platform for Oracle Internet Directory SSL Authentication	6-7
6.3.2.1	Configuring Oracle Directory Integration Platform for Oracle Internet Directory SSL Server Authentication Only Mode	6-7
6.3.2.2	Configuring Oracle Directory Integration Platform for Oracle Internet Directory SSL No Authentication Mode	6-10
6.3.3	Adding Cipher Suites Configured for Oracle Internet Directory into Oracle Directory Integration Platform	6-11
6.4	Configuring Oracle Directory Integration Platform for Oracle Internet Directory	6-13
6.5	Verifying Oracle Directory Integration Platform	6-16

## 7 Configuring Oracle Directory Server Enterprise Edition

---

7.1	Before You Configure Oracle Directory Server Enterprise Edition as the Back-End Directory	7-1
7.2	Configuring Oracle Directory Server Enterprise Edition (Non-SSL) for Oracle Directory Integration Platform	7-1
7.2.1	Installing and Configuring Oracle Directory Server Enterprise Edition	7-2
7.2.2	Installing Oracle Directory Server Enterprise Edition Plug-In	7-2
7.2.3	Creating Oracle Directory Server Enterprise Edition Suffixes	7-2
7.2.4	Enabling the Retro Change Log for Oracle Directory Server Enterprise Edition	7-3
7.2.5	Configuring the Oracle WebLogic Server Domain for Oracle Directory Integration Platform with Oracle Directory Server Enterprise Edition	7-3
7.2.5.1	Configuring Oracle Directory Integration Platform with Oracle Directory Server Enterprise Edition in an Existing WebLogic Domain	7-4
7.2.5.2	Configuring Oracle Directory Integration Platform and Oracle Directory Server Enterprise Edition in a New Oracle WebLogic Server Domain	7-5
7.2.6	Starting the Oracle WebLogic Server and the Oracle Directory Server Enterprise Edition Instance	7-7
7.2.7	Configuring Oracle Directory Integration Platform for Oracle Directory Server Enterprise Edition	7-8
7.2.8	Adding Access Control Instructions (ACIs) for Oracle Directory Server Enterprise Edition	7-9
7.3	Configuring Oracle Directory Server Enterprise Edition (SSL) for Oracle Directory Integration Platform	7-9
7.3.1	Configuring Oracle Directory Server Enterprise Edition for SSL	7-10
7.3.2	Configuring Oracle Directory Integration Platform for Oracle Directory Server Enterprise Edition SSL	7-11
7.4	Verifying Oracle Directory Integration Platform	7-13

## Part IV Synchronization Using Oracle Directory Integration Platform

---

### 8 Understanding the Oracle Directory Synchronization Service

---

8.1	Components Involved in Oracle Directory Synchronization	8-1
8.1.1	Overview of Connectors for Directory Synchronization	8-1
8.1.2	Understanding Directory Synchronization Profiles	8-2
8.2	Understanding How Synchronization Works	8-3
8.2.1	Understanding Synchronization from the Back-end Directory to a Connected Directory	8-4
8.2.2	Understanding Synchronization from a Connected Directory to the Back-end Directory	8-4

8.2.3	Understanding Directories Synchronization with Interfaces Not Supported by the Back-end Directory	8-5
-------	---	-----

## 9 Configuring Directory Synchronization

---

9.1	Registering Connectors in Oracle Directory Integration Platform	9-1
9.2	About Synchronization Profile Templates	9-2
9.3	Configure Connection Details for a Third-Party Directory	9-2
9.4	Configuring Mapping Rules	9-3
9.4.1	About Mapping Rules Attribute	9-4
9.4.2	Distinguished Name Mapping	9-5
9.4.3	About Domain Mapping Rules	9-6
9.4.4	Domain Exclusion List	9-7
9.4.5	Attribute-Level Mapping	9-8
9.4.6	Attribute Exclusion List	9-10
9.4.7	Manually Creating New Mapping Files	9-11
9.4.8	Supported Attribute Mapping Rules and Examples	9-12
9.4.9	Configuring Account Locking Synchronization	9-14
9.4.9.1	Prerequisites for Account Lockout Synchronization	9-15
9.4.9.2	Enabling Account Lockout Synchronization	9-15
9.4.10	Configuring Account Disabling Synchronization	9-16
9.4.11	Example: Mapping File for a Tagged-File Interface	9-18
9.4.12	Example: Mapping Files for an LDIF Interface	9-19
9.4.13	Updating Mapping Rules	9-20
9.4.13.1	Adding an Entry to the Mapping Rules File	9-20
9.4.13.2	Modifying an Entry in the Mapping Rules File	9-20
9.4.13.3	Deleting an Entry from the Mapping Rules File	9-20
9.5	Extending Mappings Using Custom Plug-ins	9-21
9.5.1	Writing Custom Plug-Ins	9-21
9.5.2	Understanding Mapping Plug-In Evaluation Constraints	9-22
9.5.3	Adding Mapping Plug-Ins	9-23
9.5.4	Applications of Mapping Plug-Ins	9-23
9.5.4.1	Support for New Mapping Operations	9-23
9.5.4.2	Support for Multiple Literal Values	9-24
9.5.5	Example Plug-In Usage	9-24
9.6	Configuring Matching Filters	9-25
9.6.1	Filtering Changes with an LDAP Search	9-26
9.6.2	Filtering Changes from a Change Log	9-26
9.7	Location and Naming of Files	9-27
9.8	Password Synchronization	9-27
9.8.1	Understanding Password Synchronization Mechanism	9-27
9.8.2	Configuring Password Synchronization for Oracle Unified Directory	9-31



9.8.2.1	Synchronizing Password from a Connected Directory to the Oracle Unified Directory using On-Demand Password	9-31
9.8.2.2	Synchronizing Password from the Oracle Unified Directory to a Connected Directory using Password Translate	9-33
9.8.3	Configuring Password Synchronization for Oracle Directory Server Enterprise Edition	9-35
9.8.3.1	Synchronizing Password from a Connected Directory to the Oracle Directory Server Enterprise Edition using On-Demand Password	9-35
9.8.3.2	Synchronizing Password from the Oracle Directory Server Enterprise Edition to a Connected Directory Using Password Translate	9-37
9.8.4	Configuring Password Synchronization for Oracle Internet Directory	9-38

## 10 Managing Directory Synchronization Profiles

---

10.1	Managing Synchronization Profiles Using Fusion Middleware Control	10-1
10.1.1	Creating Synchronization Profiles	10-1
10.1.2	Editing Synchronization Profiles	10-9
10.1.3	Enabling and Disabling Synchronization Profiles	10-9
10.1.4	Deleting Synchronization Profiles	10-10
10.1.5	Troubleshooting Synchronization Profiles Using DIP Tester	10-10
10.1.5.1	Running DIP Tester From the Enterprise Manager User Interface	10-11
10.1.5.2	Running DIP Tester From the WLST Command-Line Interface	10-15
10.2	Managing Synchronization Profiles Using manageSyncProfiles	10-17
10.3	Modifying the Synchronization Status Attributes	10-18
10.4	Setting Null Values in Synchronization Profiles	10-18
10.5	manageSyncProfiles Utility	10-18

## 11 Bootstrapping a Directory in Oracle Directory Integration Platform

---

11.1	Overview of Directory Bootstrapping Using syncProfileBootstrap	11-1
11.1.1	Recommended Bootstrapping Methodology	11-2
11.1.2	About How to Use a Parameter File to Bootstrap	11-2
11.1.3	Bootstrapping from an LDAP File to an LDIF File	11-2
11.1.4	Bootstrapping Using an LDIF File	11-3
11.1.4.1	Bootstrapping from an LDIF File Using Directory-Dependent Tools to Read the Source Directory	11-3
11.1.4.2	Bootstrapping from an LDIF File Using the syncProfileBootstrap Command to Load Data into the Back-end Directory	11-4
11.1.5	Bootstrapping Directly Using the Default Integration Profile	11-5
11.2	Bootstrapping in SSL Mode	11-5
11.3	syncProfileBootstrap Utility	11-7

## 12 Synchronizing with Tables in Oracle Database

---

12.1	Overview of the Additional Configuration Information File	12-2
12.1.1	Preparing the Additional Configuration Information File	12-2
12.1.2	Understanding the Format of the Additional Configuration Information File	12-2
12.1.3	Understanding the DBReader.cfg.master Configuration File	12-3
12.1.4	Understanding the dbexport.cfg.master Configuration File	12-4
12.2	Updating the Configuration File	12-5
12.2.1	The DBReader.map.master Configuration File	12-5
12.2.2	The dbexport.map.master Configuration File	12-6
12.3	Preparing the Directory Integration Profile	12-6
12.4	Example: Synchronizing a Relational Database Table to the Back-end Directory	12-7
12.4.1	About the Example Scenario	12-7
12.4.2	Configure the Additional Configuration Information File	12-8
12.4.3	Configure the Mapping File	12-8
12.4.4	Configure the Directory Integration Profile	12-9
12.4.5	Uploading the Additional Configuration Information and Mapping Files	12-10
12.4.6	Understanding Synchronization Process for Relational Database Table with the Oracle Back-End Directory	12-10
12.4.7	Observations About the Synchronizing a Relational Database Table to the Back-End Directory Example	12-11

## 13 Synchronizing with Oracle Human Resources

---

13.1	Introduction to Synchronization with Oracle Human Resources	13-1
13.2	What are the Data you can Import from Oracle Human Resources?	13-2
13.3	Managing Synchronization Between Oracle Human Resources and the Oracle Back-end Directory	13-3
13.3.1	Configure a Directory Integration Profile for the Oracle Human Resources Connector	13-4
13.3.2	About Oracle Human Resources Attributes to be Synchronized with the Oracle Back-end Directory	13-5
13.3.2.1	Configure the List of Attributes to be Synchronized with the Oracle Back-end Directory	13-6
13.3.2.2	Modifying Additional Oracle Human Resources Attributes for Synchronization	13-7
13.3.2.3	Excluding Oracle Human Resources Attributes from Synchronization	13-8
13.3.2.4	Configure a SQL SELECT Statement in the Configuration File to Support Complex Selection Criteria	13-8
13.3.3	About Mapping Rules for the Oracle Human Resources Connector	13-9
13.3.4	Setting Up Synchronization from Oracle Human Resources to the Oracle Back-end Directory	13-9

13.4	Understanding the Synchronization Process between Oracle Human Resources and the Oracle Back-End directory	13-10
13.5	Bootstrapping the Oracle Back-end Directory from Oracle Human Resources	13-11

## 14 Synchronizing with Third-Party Metadirectory Solutions

---

14.1	About Change Logs	14-1
14.2	Enabling Third-Party Metadirectory Solutions to Synchronize with the Oracle Back-end Directory	14-2
14.2.1	Performing Initial Bootstrapping	14-2
14.2.2	Understand How to Change Subscription Object in the Oracle Back-end Directory for the Third-Party Metadirectory Solution	14-3
14.2.2.1	About the Change Subscription Object	14-3
14.2.2.2	Creating a Change Subscription Object	14-3
14.3	Understanding Synchronization Process	14-4
14.3.1	Retrieving Changes the First Time from the Oracle Back-end Directory	14-4
14.3.2	Updating the orclLastAppliedChangeNumber Attribute in the Oracle Back-end Directory	14-5
14.4	Disabling and Deleting Change Subscription Objects	14-5
14.4.1	Disabling a Change Subscription Object	14-6
14.4.2	Deleting a Change Subscription Object	14-6

## Part V Provisioning with the Oracle Directory Integration Platform

---

### 15 Understanding the Oracle Directory Integration Platform for Provisioning

---

15.1	What Is Provisioning?	15-1
15.2	Understanding Oracle Provisioning Products	15-2
15.3	Oracle Directory Integration Platform Service Components	15-3
15.4	Understanding Provisioning Concepts	15-3
15.4.1	About Synchronous Provisioning	15-4
15.4.2	About Asynchronous Provisioning	15-5
15.4.3	About Provisioning Data Flow	15-6
15.5	Overview of Provisioning Methodologies	15-7
15.5.1	Understanding Provisioning Users that are Synchronized from an External Source	15-7
15.5.2	Understanding Provisioning Users Created with Command-Line LDAP Tools	15-8
15.5.3	Bulk Provisioning Using the provProfileBulkProv Tool	15-8
15.5.4	On-Demand Provisioning	15-8
15.5.5	Application Bootstrapping	15-8

15.6	About Organization of User Profiles in the Oracle Back-End Directory	15-8
15.6.1	Understanding Organization of Provisioning Entries in the Directory Information Tree	15-8
15.6.2	Understanding User Provisioning Status	15-10
15.6.2.1	Provisioning Status in Oracle Back-end Directory	15-10
15.6.2.2	Provisioning Status Transitions	15-12
15.6.2.3	Understanding Upgrade and Coexistence Provisioning Status	15-13
15.6.2.4	Understanding Provisioning Status and Exception Handling	15-14
15.7	Understanding Provisioning Flow	15-14
15.7.1	Viewing and Editing Provisioning Profiles Using Fusion Middleware Control	15-14
15.7.2	Provisioning User from an External Source	15-15
15.8	Understanding the Delegation of Administrative Privileges	15-16
15.9	provProfileBulkProv utility	15-17

## 16 Deploying Provisioning-Integrated Applications

---

16.1	Deploying Provisioning-Integrated Applications	16-1
16.2	About manageProvProfiles Command	16-2
16.3	Tasks and Examples for manageProvProfiles	16-3
16.3.1	Creating a Provisioning Profile	16-3
16.3.2	Modifying a Provisioning Profile	16-4
16.3.3	Deleting a Provisioning Profile	16-4
16.3.4	Disabling a Provisioning Profile	16-4
16.3.5	Re-enabling a Disabled Provisioning Profile	16-5
16.4	Registering Applications for Provisioning	16-5
16.5	Understanding How to Configure Application Provisioning Properties	16-9
16.6	manageProvProfiles Command	16-10

## 17 Understanding the Oracle Provisioning Event Engine

---

17.1	What Are the Oracle Provisioning Events?	17-1
17.2	Working with the Oracle Provisioning Event Engine	17-1
17.2.1	Create Custom Event Object Definitions	17-2
17.2.2	Define Custom Event Generation Rules	17-2

## 18 Integration of Provisioning Data with Oracle E-Business Suite

---

### 19 Connected Directory Integration Concepts and Considerations

---

19.1	Concepts and Architecture of Connected Directory Integration	19-1
19.1.1	Oracle Identity Management Components for Integrating with Other Directories	19-2
19.1.1.1	Understanding Oracle Back-end Directory	19-2
19.1.1.2	Understanding Oracle Directory Integration Platform	19-3
19.1.1.3	Understanding Oracle Directory Services Manager	19-3
19.1.1.4	Understanding Delegated Authentication	19-3
19.1.2	Oracle Back-end Directory Schema Elements for Synchronizing with Connected Directories	19-4
19.1.3	Directory Information Tree in an Integration with a Connected Directory	19-4
19.1.3.1	About Realms in Oracle Internet Directory	19-5
19.1.3.2	About Suffixes in Oracle Unified Directory and Oracle Directory Server Enterprise Edition	19-7
19.1.3.3	Understanding How to Plan the Deployment	19-7
19.1.3.4	Example: Integration with a Single Connected Directory Domain	19-8
19.2	Planning Your Integration Environment	19-9
19.2.1	Preliminary Considerations for Integrating with a Connected Directory	19-9
19.2.2	Choose the Directory for the Central Enterprise Directory	19-10
19.2.2.1	Overview of Scenario 1: Oracle Directory as the Central Enterprise Directory	19-10
19.2.2.2	Overview of Scenario 2: A Directory Other than the Oracle Directory is the Central Enterprise Directory	19-13
19.2.3	Understanding How to Customize the LDAP Schema	19-15
19.2.4	About Choosing Where to Store Passwords	19-15
19.2.4.1	Understanding the Advantages and Disadvantages of Storing the Password in One Directory	19-16
19.2.4.2	Understanding the Advantages and Disadvantages of Storing Passwords in Both Directories	19-16
19.2.5	About Choosing the Structure of the Directory Information Tree	19-18
19.2.5.1	Create Identical DIT Structures on Both Directories	19-18
19.2.5.2	Distinguished Name Mapping and Limitations	19-19
19.2.6	Attribute for the Login Name	19-20
19.2.7	Selecting the User Search Base	19-21
19.2.8	What Group Search Base to Select?	19-21
19.2.9	Guidelines to Address Security Concerns	19-22
19.2.10	About How to Administer Your Deployment with Oracle Access Manager	19-22
19.3	Microsoft Active Directory Integration Concepts	19-22

19.3.1	Understanding How to Synchronize from Microsoft Active Directory to the Oracle Back-end Directory	19-23
19.3.2	Requirement for Using WebDAV Protocol	19-25
19.3.3	Oracle Back-end Directory Schema Elements for Microsoft Active Directory	19-25
19.3.4	About Integration with Multiple Microsoft Active Directory Domain Controllers	19-26
19.3.5	Synchronizing with a Multiple-Domain Microsoft Active Directory Environment	19-27
19.3.5.1	About Configuration Required for Importing from Microsoft Active Directory to the Oracle Back-end Directory	19-27
19.3.5.2	About Configuration Required for Importing from Microsoft Active Directory Lightweight Directory Service to the Oracle Back-end Directory	19-28
19.3.5.3	About Configuration Required for Exporting from the Oracle Back-end Directory to Microsoft Active Directory	19-28
19.3.5.4	Example: Integration with Multiple Connected Directory Domains	19-28
19.3.6	Understanding Foreign Security Principals	19-29
19.4	Oracle Directory Server Enterprise Edition (Sun Java System Directory Server) Integration Concepts	19-30
19.4.1	Understanding Synchronization from Oracle Directory Server Enterprise Edition to Oracle Directory Integration Platform	19-30
19.4.2	Understanding Oracle Internet Directory Schema Elements for Oracle Directory Server Enterprise Edition (Sun Java System Directory Server)	19-31
19.5	IBM Tivoli Directory Server Integration Concepts	19-31
19.5.1	About Changes to Directory Objects in IBM Tivoli Directory Server	19-31
19.5.2	Understanding Oracle Back-end Directory Schema Elements for IBM Tivoli Directory Server	19-32
19.6	Novell eDirectory and OpenLDAP Integration Concepts	19-32
19.6.1	About Synchronization from Novell eDirectory or OpenLDAP to the Oracle Back-end Directory	19-33
19.6.2	Understanding Oracle Back-end Directory Schema Elements for Novell eDirectory	19-33
19.6.3	Understanding Oracle Back-end Directory Schema Elements for OpenLDAP	19-34
19.7	Limitations of Connected Directory Integration in Oracle Directory Integration Platform	19-35

## 20 Configuring Synchronization with a Connected Directory

---

20.1	Verifying Synchronization Requirements	20-1
20.2	Creating Import and Export Synchronization Profiles Using expressSyncSetup	20-3
20.2.1	Understanding the expressSyncSetup Command	20-4

20.2.2	Verifying Users and Groups Synchronization in the Connected Directory	20-5
20.3	Configuring Advanced Integration Options	20-6
20.3.1	Configuring the Realm	20-7
20.3.2	Customizing Access Control Lists	20-8
20.3.2.1	Customizing ACLs for Import Profiles	20-8
20.3.2.2	Customizing ACLs for Export Profiles	20-10
20.3.3	Customizing Mapping Rules	20-11
20.3.4	Configuring the Connected Directory Connector for Synchronization in SSL Mode	20-12
20.3.5	Enable Password Synchronization from the Oracle Back-end Directory to a Connected Directory	20-15
20.3.6	Configuring External Authentication Plug-ins	20-15
20.3.7	Configuring External Authentication Against Multiple Domains	20-17
20.4	Writing Custom Synchronization Connectors	20-20
20.4.1	Writing Inbound Connectors	20-20
20.4.2	Writing an Outbound Connectors	20-21
20.5	Sample Reader for Inbound Connectors	20-22
20.6	Sample Writer for Outbound Connectors	20-27
20.7	expressSyncSetup command	20-32

## 21 Integrating with Microsoft Active Directory

---

21.1	Verify Synchronization Requirements for Microsoft Active Directory	21-1
21.2	Configuring Basic Synchronization with Microsoft Active Directory	21-2
21.3	Configuring Advanced Integration with Microsoft Active Directory	21-2
21.3.1	Understanding How to Plan Integration with Microsoft Active Directory	21-4
21.3.2	Configure the Realm for Microsoft Active Directory	21-4
21.3.3	Customizing the Search Filter to Retrieve Information from Microsoft Active Directory	21-4
21.3.4	Understanding How to Customize the ACLs for Microsoft Active Directory	21-5
21.3.5	Customize Attribute Mappings for Integrating with Microsoft Active Directory	21-5
21.3.6	Synchronizing with Multiple Microsoft Active Directory Domains	21-6
21.3.7	About How to Synchronize Deletions from Microsoft Active Directory	21-7
21.3.8	About Synchronization in SSL Mode	21-8
21.3.9	Synchronizing Passwords from the Oracle back-End Directory to Microsoft Active Directory	21-8
21.3.10	About the Microsoft Active Directory External Authentication Plug-in Configuration	21-8
21.3.11	Perform Post-Configuration and Administrative Tasks	21-8
21.4	Using DirSync Change Tracking for Import Operations	21-8

21.5	Configuring Synchronization of Microsoft Active Directory Foreign Security Principal References with an Oracle Back-End Directory	21-9
21.6	Switching to a Different Microsoft Active Directory Domain Controller in the Same Domain	21-12
21.6.1	Changing the Microsoft Active Directory Domain Controller by Using the USN-Changed Approach	21-12
21.6.2	Changing the Microsoft Active Directory Domain Controller by Using the DirSync Approach	21-13
21.7	About Configuration for Microsoft Active Directory Connector with Microsoft Active Directory Lightweight Directory Service	21-13
21.8	Configuring the Microsoft Active Directory Connector for Microsoft Exchange Server	21-13
21.8.1	Enabling Microsoft Exchange User Synchronization Using the Fusion Middleware Control	21-14
21.8.2	Enabling Microsoft Exchange User Synchronization From the Command Line	21-15

## 22 Deploying the Oracle Password Filter for Microsoft Active Directory

---

22.1	Overview of the Oracle Password Filter for Microsoft Active Directory	22-1
22.1.1	What is the Oracle Password Filter for Microsoft Active Directory?	22-2
22.1.2	Learn How the Oracle Password Filter for Microsoft Active Directory Work?	22-3
22.1.2.1	Understanding How Clear Text Password Changes are Captured	22-3
22.1.2.2	Understanding How Password Changes are Stored when the Oracle Back-end Directory is Unavailable	22-4
22.1.2.3	About Delay in Password Synchronization Until Microsoft Active Directory Users are Synchronized with Oracle Back-end Directory	22-4
22.1.2.4	Understanding Password Bootstrapping	22-4
22.1.3	Deploying the Oracle Password Filter for Microsoft Active Directory?	22-4
22.2	Understanding How to Configure and Test Oracle Back-end Directory with SSL Server-Side Authentication	22-5
22.3	Importing a Trusted Certificate into a Microsoft Active Directory Domain Controller	22-6
22.4	Testing SSL/TLS Communication Between Oracle Back-end directory and Microsoft Active Directory	22-7
22.5	Installing and Reconfiguring the Oracle Password Filter for Microsoft Active Directory	22-8
22.5.1	Prerequisites to Install or Reconfigure the Oracle Password Filter for Microsoft Active Directory	22-8
22.5.2	Installing the Oracle Password Filter for Microsoft Active Directory	22-10
22.5.3	Reconfiguring the Oracle Password Filter for Microsoft Active Directory	22-13
22.6	Removing the Oracle Password Filter for Microsoft Active Directory	22-15



## 23 Integrating with Oracle Directory Server Enterprise Edition (Connected Directory)

---

23.1	Verifying Synchronization Requirements for Oracle Directory Server Enterprise Edition	23-1
23.2	Configuring Basic Synchronization with Oracle Directory Server Enterprise Edition	23-2
23.3	Configuring Advanced Integration with Oracle Directory Server Enterprise Edition	23-2
23.3.1	Understanding How to Plan Integration with Oracle Directory Server Enterprise Edition	23-3
23.3.2	Configure the Realm for Oracle Directory Server Enterprise Edition	23-3
23.3.3	Understanding How to Customize the ACLs for Oracle Directory Server Enterprise Edition	23-3
23.3.4	Customize Attribute Mappings for Oracle Directory Server Enterprise Edition	23-3
23.3.5	About How to Customize the Oracle Directory Server Enterprise Edition Connector to Synchronize Deletions	23-4
23.3.6	Understanding How to Synchronize Passwords for Oracle Directory Server Enterprise Edition	23-4
23.3.7	Synchronizing in SSL Mode	23-4
23.3.8	Perform Post-Configuration and Administrative Tasks	23-4

## 24 Integrating with IBM Tivoli Directory Server

---

24.1	Verifying Synchronization Requirements for IBM Tivoli Directory Server	24-1
24.2	Configuring Basic Synchronization with IBM Tivoli Directory Server	24-2
24.3	Configuring Advanced Integration with IBM Tivoli Directory Server	24-2
24.3.1	Understanding How to Plan Integration with IBM Tivoli Directory	24-3
24.3.2	Configure the Realm for IBM Tivoli Directory	24-3
24.3.3	Understanding How to Customize the ACLs for IBM Tivoli Directory	24-3
24.3.4	Customize Attribute Mappings for IBM Tivoli Directory	24-3
24.3.5	Customizing the IBM Tivoli Directory Server Connector to Synchronize Deletions	24-4
24.3.6	Synchronize Passwords for IBM Tivoli Directory	24-4
24.3.7	Understand How to Synchronize IBM Tivoli Directory in SSL Mode	24-4
24.3.8	Configuring the IBM Tivoli Directory Server External Authentication Plug-in	24-5
24.3.9	Perform Post-Configuration and Administrative Tasks	24-6

## 25 Integrating with Novell eDirectory or OpenLDAP

---

25.1	Verify Synchronization Requirements for Novell eDirectory or OpenLDAP	25-1
25.2	Configuring Basic Synchronization with Novell eDirectory or OpenLDAP	25-2

25.3	Synchronizing Multiple Profiles from eDirectory or OpenLDAP to One Oracle Back-end Directory Container	25-2
25.4	Configuring Advanced Integration with Novell eDirectory or OpenLDAP	25-2
25.4.1	Understanding How to Plan Integration with Novell eDirectory or OpenLDAP	25-3
25.4.2	About Realm Configuration for Novell eDirectory or OpenLDAP	25-3
25.4.3	Customize the Search Filter to Retrieve Information from Novell eDirectory or OpenLDAP	25-4
25.4.4	Understanding the ACLs Customization for Novell eDirectory or OpenLDAP	25-4
25.4.5	Customize Attribute Mappings for Novell eDirectory	25-5
25.4.6	Customizing the Novell eDirectory or OpenLDAP Connector to Synchronize Deletions	25-5
25.4.7	About How to Define a Reconciliation Rule?	25-6
25.4.8	Reconciliation Rules Used to Synchronize Deletions?	25-6
25.4.9	Synchronization Parameters for the Advanced Configuration Information Attribute	25-7
25.4.10	Configuring the OpenLDAP Connector to Synchronize Passwords	25-9
25.4.11	Synchronize the Novell eDirectory or OpenLDAP connector in SSL Mode	25-10
25.4.12	Configure the Novell eDirectory or OpenLDAP External Authentication Plug-in	25-10
25.4.13	Perform Post-Configuration and Administrative Tasks	25-10

## 26 Managing Integration with a Connected Directory

---

26.1	Performing Tasks After Configuring with a Connected Directory	26-1
26.2	Typical Management of Integration with a Connected Directory	26-1
26.2.1	Overview of Management Task for after Integration with a Connected Directory	26-2
26.2.2	Bootstrapping Data Between Directories	26-2
26.2.3	Managing a Third-Party Directory External Authentication Plug-in	26-3
26.2.3.1	Deleting a Third-Party Directory External Authentication Plug-in	26-3
26.2.3.2	Disabling a Third-Party External Authentication Plug-in	26-3
26.2.3.3	Re-enabling a Third-Party External Authentication Plug-in	26-4
26.2.4	Granting Password Privilege for Oracle Unified Directory Acting As Connected Directory	26-4

## Part VII Appendixes

---

<b>A</b>	<b>Example Properties File for Synchronization Profiles</b>	
A.1	Example Profile Properties File	A-1
<b>B</b>	<b>Case Study: A Deployment of Oracle Directory Integration Platform</b>	
B.1	Components in the MyCompany Enterprise	B-1
B.2	Requirements of the MyCompany Enterprise	B-1
B.3	Overall Deployment in the MyCompany Enterprise	B-1
B.4	User Creation and Provisioning in the MyCompany Enterprise	B-2
B.5	Modification of User Properties in the MyCompany Enterprise	B-3
B.6	Deletion of Users in the MyCompany Enterprise	B-4
<b>C</b>	<b>Starting and Stopping the Oracle Stack</b>	
C.1	Starting the Stack	C-1
C.2	Stopping the Stack	C-2
C.3	About How to Start or Stop the Oracle Unified Directory Server	C-3
C.4	About How to Start or Stop the Oracle Internet Directory Server	C-3
C.5	About How to Start, Stop, or Restart a Oracle Directory Server Enterprise Edition Instance	C-4
<b>D</b>	<b>Troubleshooting the Oracle Directory Integration Platform</b>	
D.1	Checklist for Troubleshooting Oracle Directory Integration Platform	D-1
D.2	General Issues	D-2
D.2.1	LDIF Files That Contain Non-ASCII Characters Will Cause the testProfile Command Option to Fail if the LDIF File has Native Encoding	D-3
D.2.2	Some Changes May Not Get Synchronized Due to Race Condition in Heavily-Loaded Source Directory	D-3
D.2.3	Synchronization Continues After Stopping Oracle Directory Integration Platform	D-4
D.2.4	Synchronization Continues After Disabling Synchronization Profiles for Oracle Unified Directory (Back-End Directory)	D-4
D.2.5	Synchronization of Deleted Objects Fails	D-4
D.3	Configuration Issues	D-4
D.3.1	You may Need to Restart the Directory Integration Platform After Running dipConfigurator Against Oracle Unified Directory	D-5
D.3.2	When Configuring a Profile, you may Need to Scroll Past a Section of Whitespace to View Mapping Rules	D-5
D.4	Problems and Solutions	D-5
D.4.1	Provisioning Errors and Problems	D-5
D.4.2	Synchronization Errors and Problems	D-8
D.4.3	Novell eDirectory and OpenLDAP Synchronization Errors and Problems	D-11

D.4.4	Oracle Password Filter for Microsoft Active Directory Errors and Problems	D-13
D.5	Troubleshooting Synchronization	D-16
D.5.1	Oracle Directory Integration Platform Synchronization Process Flow	D-17
D.5.1.1	Oracle Directory Integration Platform Synchronization Process Flow for an Import Profile	D-17
D.5.1.2	Oracle Directory Integration Platform Synchronization Process Flow for an Export Profile	D-18
D.5.2	Understanding Synchronization Profile Registration	D-18
D.5.3	Understanding the diagnostic.log File	D-19
D.6	Troubleshooting Integration with Microsoft Active Directory	D-23
D.6.1	Debugging Windows Native Authentication	D-23
D.6.2	Synchronizing Changes Following a Period when the Oracle Back-end Directory is Unavailable	D-24
D.7	Troubleshooting SSL/TLS	D-26
D.8	Need More Help?	D-27

## Glossary

---

## List of Examples

---

9-1	Example of Distinguished Name Mapping	9-5
9-2	Example of One-to-One Distinguished Name Mapping	9-5
9-3	Example Map File Using DomainExclusionList and AttributeExclusionList Headers	9-10
10-1	General	10-2
10-2	Mapping	10-3
10-3	Filtering	10-5
10-4	Advanced	10-6
10-5	Test Mode	10-11
10-6	Test Params	10-13
10-7	View Test Data	10-13
10-8	Test Options	10-14
10-9	Review Options and Test Output	10-14
16-1	Oracle Internet Directory	16-6
16-2	Oracle Unified Directory	16-7
16-3	Oracle Internet Directory	16-7
16-4	Oracle Unified Directory	16-7
21-1	Attribute-Level Mapping for the User Object in Microsoft Active Directory	21-6
21-2	Attribute-Level Mapping for the Group Object in Microsoft Active Directory	21-6
21-3	How Foreign Key References Are Resolved	21-11
23-1	Attribute-Level Mapping for the User Object in Oracle Directory Server Enterprise Edition	23-3
23-2	Attribute-Level Mapping for the Group Object in Oracle Directory Server Enterprise Edition	23-4
24-1	Attribute-Level Mapping for the User Object in IBM Tivoli Directory Server	24-3
24-2	Attribute-Level Mapping for the Group Object in IBM Tivoli Directory Server	24-3
25-1	Attribute-Level Mapping for the User Object in Novell eDirectory or OpenLDAP	25-5
25-2	Attribute-Level Mapping for the Group Object in Novell eDirectory or OpenLDAP	25-5

## List of Figures

---

1-1	Example of an Oracle Directory Integration Platform Environment	1-7
1-2	Interactions of the Oracle Directory Integration Platform Synchronization Service	1-9
1-3	Interactions of the Oracle Directory Integration Platform Provisioning Service	1-11
15-1	Synchronous Provisioning from Command-Line LDAP Tools	15-5
15-2	Asynchronous Provisioning using Command-Line LDAP Tools	15-6
15-3	Provisioning Data Flow	15-7
15-4	Oracle Back-End Directory Base User and Application-Specific Attributes	15-10
15-5	Valid Provisioning Status Transitions	15-13
19-1	The Default Identity Management Realm	19-6
19-2	Default DIT Structures in Oracle Internet Directory and a Connected Directory When Both Directory Hosts Are Under the Domain us.MyCompany.com	19-8
19-3	Interaction Among Components with Oracle Directory as the Central Enterprise Directory	19-12
19-4	Interaction of Components with a Third-Party Directory as the Central Enterprise Directory in a Delegated Authentication Deployment	19-14
19-5	Mapping Between the Oracle Back-end Directory and a Forest in Microsoft Active Directory	19-26
19-6	Example of a Mapping Between the Oracle Back-end Directory and Multiple Domains in Microsoft Active Directory	19-29
B-1	Example of Oracle Directory Integration Platform in the MyCompany Deployment	B-2
B-2	User Creation and Provisioning	B-3
B-3	Modification of User Properties	B-4
B-4	Deletion of Users from the Corporate Human Resources	B-5

## List of Tables

---

1-1	Directory Synchronization and Provisioning Integration Distinctions	1-5
2-1	Out-of-Box Cipher Suites	2-3
2-2	createCred Commands	2-12
4-1	dipStatus utility Arguments	4-13
4-2	dipStatus utility Arguments	4-14
5-1	dipConfigurator Properties for Oracle Unified Directory	5-10
6-1	dipConfigurator Properties for Oracle Internet Directory	6-14
7-1	dipConfigurator Properties for Oracle Directory Server Enterprise Edition	7-8
9-1	Connection Detail Properties	9-3
9-2	Domain Rule Components	9-5
9-3	Components in Attribute Rules	9-8
9-4	Mapping Attribute for Account Disabling Synchronization	9-16
9-5	Location and Names of Files	9-27
9-6	Password Synchronization Mechanism	9-29
9-7	setupPlugin Properties	9-31
9-8	setupPlugin Properties	9-35
10-1	Synchronization Profile Properties, Basic Properties	10-12
10-2	Synchronization Profile Properties, Source Details or Destination Details	10-12
10-3	Synchronization Profile Properties, Advanced	10-13
11-1	syncProfileBootstrap utility Arguments	11-8
12-1	Directory Integration Profile for TESTDBIMPORT	12-9
13-1	Tables in Oracle Human Resources Schema	13-2
13-2	Fields in the Oracle Human Resources User Interface	13-2
13-3	Attributes Specific to Oracle Human Resources Connector Integration Profile	13-4
13-4	Oracle Human Resources Attributes Synchronized with the Oracle Back-end Directory by Default	13-6
15-1	Provisioning Status in the Back-end Directory	15-10
15-2	Valid Provisioning Status Transitions in the Back-end Directory	15-12
15-3	Provisioning Profile Fields	15-15
15-4	dipStatus utility Arguments	15-17
16-1	Common Privileged Groups in Oracle Internet Directory	16-8
17-1	Event Object Properties	17-2
17-2	Predefined Event Objects	17-2
17-3	Supported Event Definitions	17-3
19-1	Typical Requirements with Oracle Directory as the Central Enterprise Directory	19-10

19-2	Typical Requirements if a Connected Directory is the Central Enterprise Directory	19-13
19-3	Comparing the DirSync Approach to the USN-Changed Approach	19-23
19-4	Oracle Back-end Directory Schema Elements for Microsoft Active Directory	19-25
19-5	Oracle Back-end Directory Schema Elements for IBM Tivoli Directory Server	19-32
19-6	Oracle Back-end Directory Schema Elements for Novell eDirectory	19-33
19-7	Oracle Internet Directory Schema Elements for OpenLDAP	19-34
20-1	Supported Values for sslmode in connectedDirectoryURL Attribute	20-14
20-2	Distinguished Names of External Authentication Plug-ins	20-18
20-3	dipStatus utility Arguments	20-33
22-1	Oracle Password Filter Configuration Parameters for Microsoft Active Directory	22-8
22-2	Oracle Password Filter Configuration Parameters for Oracle Back-end Directory	22-9
25-1	Novell eDirectory and OpenLDAP Synchronization Parameters for the Advanced Configuration Information Attribute	25-7



# Preface

*Oracle Identity Management* describes how to use Oracle Directory Integration Platform to integrate applications and directories—including third-party LDAP directories—with Oracle Unified Directory, Oracle Internet Directory, or Oracle Directory Server Enterprise Edition.

## Audience

*Oracle Identity Management* is intended for anyone who needs to integrate Oracle Unified Directory, Oracle Internet Directory, or Oracle Directory Server Enterprise Edition 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 that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Related Documentation

For more information, see:

- Online help available through Oracle Directory Services Manager and Oracle Enterprise Manager.
- The Oracle WebLogic Server, Oracle Database, and Oracle Identity Management documentation sets, especially:
  - *Oracle Fusion Middleware Installing Oracle Unified Directory*
  - *Oracle Fusion Middleware Administering Oracle Unified Directory*
  - *Oracle Fusion Middleware Developing Plug-Ins for Oracle Unified Directory*
  - *Oracle Fusion Middleware Configuration Reference for Oracle Unified Directory*

- *Oracle Fusion Middleware Java API Reference for Oracle Unified Directory*
- *Oracle Fusion Middleware Installing and Configuring Oracle Internet Directory*
- *Oracle Fusion Middleware Administering Oracle Internet Directory*
- *Oracle Fusion Middleware Application Developer's Guide for Oracle Identity Management*
- *Oracle Fusion Middleware Reference for Oracle Identity Management*
- *Oracle Fusion Middleware Installation Guide for Oracle Directory Server Enterprise Edition*
- *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*
- *Oracle Fusion Middleware Developer's Guide for Oracle Directory Server Enterprise Edition*
- *Oracle Fusion Middleware High Availability Guide*
- *Oracle Fusion Middleware Administering Oracle Fusion Middleware*
- *Oracle Fusion Middleware Securing Applications with Oracle Platform Security Services*
- *Oracle Fusion Middleware WLST Command Reference for WebLogic Server*
- *Oracle Fusion Middleware Understanding the 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
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
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?

This preface introduces the new and changed features of Oracle Directory Integration Platform since the previous release, and provides pointers to additional information.

## Topics:

- [New and Changed Features for 12c \(12.2.1.3.0\)](#)
- [New and Deprecated Terminology for Oracle Directory Integration Platform 12c](#)

## New and Changed Features for 12c (12.2.1.3.0)

Oracle Directory Integration Platform 12c supports Transparent Layer Security and Cipher Suites. See [Transport Layer Security Protocol and Cipher Suites](#).

## New and Deprecated Terminology for Oracle Directory Integration Platform 12c

Terminology Used in 11g	Terminology Used in 12c
Middleware home <i>MW_HOME</i> In 11g the Middleware home is a container for the Oracle WebLogic Server home, and, optionally, one Oracle Common home and one or more Oracle homes. <b>This term is eliminated in 12c.</b>	Oracle home <i>ORACLE_HOME</i> The Oracle home that is created for all the Oracle Fusion Middleware products on a host computer. It includes binary and library files, the Oracle common directory and the individual product directories for each Oracle Fusion Middleware product you install.
Oracle home <i>PRODUCT_ORACLE_HOME</i>	Product directory <i>PRODUCT_DIR</i> The product directories within the Middleware home are no longer Oracle homes. They are simply directories within the Oracle home that are created for all the Oracle Fusion Middleware products. Most Oracle Fusion Middleware components should be installed in the same Oracle home. The names of the product homes are predefined and can no longer be modified by the user during the installation.

Terminology Used in 11g	Terminology Used in 12c
<p>Oracle instance  <b>This term is eliminated in 12c.</b></p>	<p>The installer will no longer create a separate instance directory for system components, such as Oracle HTTP Server. Instead, you can use the Fusion Middleware Configuration Wizard to configure your system components, just as you do for Java components. Instance information about each system component will be stored in the domain home.</p>
<p>Oracle Fusion Middleware farm  <b>This term is eliminated in 12c.</b></p>	<p>The term "farm" is no longer necessary for 12c. It was used in 11g to refer to a container for a WLS domain and its associated system component instances when presented in Fusion Middleware Control.</p>

See What Are the Key Oracle Fusion Middleware Directories? in *Understanding Oracle Fusion Middleware*.

# Part I

## Getting Started with Oracle Directory Integration Platform

Before you begin working with Oracle Directory Integration Platform, you need to understand the concepts, components, architecture, and security features.

Part I includes the following chapters:

- [Introduction to Oracle Directory Integration Platform](#)
- [Security Features in Oracle Directory Integration Platform](#)

# 1

## Introduction to Oracle Directory Integration Platform

This chapter introduces Oracle Directory Integration Platform, its components, structure, and administration tools.

### Topics:

- [Why Oracle Directory Integration Platform?](#)
- [Oracle Directory Integration Platform Installation Options](#)
- [Understanding the Differences Between Synchronization and Provisioning](#)
- [Understanding Components Involved in Oracle Directory Integration Platform Integration](#)

### See Also:

[Case Study: A Deployment of Oracle Directory Integration Platform](#) for an example on how you can deploy Oracle Directory Integration Platform integration.

## 1.1 Why Oracle Directory Integration Platform?

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 Unified Directory, Oracle Internet 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 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.
- Synchronize the password in Microsoft Active Directory with that in Oracle Unified Directory or Oracle Directory Server Enterprise Edition.

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 2012, 2012 R2, and 2016
- 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 11g Release 1 (11.1.1.7.0)
- Novell eDirectory 8.8
- OpenLDAP 2.4
- IBM Tivoli Directory Server 6.3
- Oracle Unified Directory
- Oracle Internet Directory

 **Note:**

Oracle Internet Directory 10g is not certified/supported as the Oracle back-end directory or connected directory for Oracle Directory Integration Platform 12c.

 **See Also:**

For a complete list of supported directories, refer to the [Oracle Fusion Middleware Supported System Configurations](#) certification matrix.

Scroll down to System Requirements and Supported Platforms for Oracle Fusion Middleware 12c (12.2.1.3.0) and then click the [xls](#) link to view the certification matrix.

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 Unified Directory, Oracle Internet Directory, or Oracle Directory Server Enterprise Edition.

 **See Also:**

- [Synchronizing with Oracle Human Resources](#)
- [Integrating with Microsoft Active Directory](#)
- [Integrating with Oracle Directory Server Enterprise Edition \(Connected Directory\)](#)



## 1.2 Oracle Directory Integration Platform 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 Directory Integration Platform, Oracle Directory Services Manager (ODSM), or Fusion Middleware Control) in a dedicated Oracle WebLogic domain on a dedicated server.

To install a standalone Oracle Directory Integration Platform instance, you first need to install an Oracle Unified Directory or Oracle Internet 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:**

Configuring Oracle Directory Integration Platform in *Installing and Configuring Oracle Internet Directory* for complete information about installing Oracle Directory Integration Platform.

## 1.3 Understanding the Differences Between Synchronization and Provisioning

Before you begin working with Oracle Directory Integration Platform, you need to understand the differences between synchronization and provisioning.

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 back-end 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:

- [About Synchronization](#)
- [About Provisioning](#)
- [How Synchronization and Provisioning Differ](#)

## 1.3.1 About 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.

You can configure the following directories as the back-end directory for Oracle Directory Integration Platform synchronization:

- Oracle Unified Directory
- Oracle Internet Directory
- Oracle Directory Server Enterprise Edition

 **See:**

[Synchronization Using Oracle Directory Integration Platform](#)

## 1.3.2 About 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 `manageProvProfiles` utility.



### See Also:

[Provisioning with the Oracle Directory Integration Platform](#)

## 1.3.3 How Synchronization and Provisioning Differ

Synchronization and provisioning have important operational differences.

[Table 1-1](#) discusses the differences.

**Table 1-1 Directory Synchronization and Provisioning Integration Distinctions**

Consideration	Directory Synchronization	Provisioning Integration
<b>The time for action</b>	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.
<b>Communication direction</b>	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.
<b>Type of data</b>	Any data in a directory.	Restricted to provisioned users and groups.

**Table 1-1 (Cont.) Directory Synchronization and Provisioning Integration Distinctions**

Consideration	Directory Synchronization	Provisioning Integration
<b>Examples</b>	Oracle Human Resources Oracle Directory Server Enterprise Edition Oracle Unified Directory Oracle Internet Directory Microsoft Active Directory Novell eDirectory OpenLDAP IBM Tivoli Directory Server Oracle Database	Oracle Portal

## 1.4 Understanding Components Involved in Oracle Directory Integration Platform Integration

Oracle Directory Integration Platform integration includes the Oracle Back-End Directory and the Oracle Directory Integration Platform.

### Topics:

- [About Oracle Back-End Directory](#)
- [About Oracle Directory Integration Platform](#)

### 1.4.1 About Oracle Back-End Directory

Either Oracle Unified Directory, Oracle Internet 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 connected 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. In this case, the Oracle Internet Directory acting as the back-end directory uses an external authentication plug-in that authenticates users against the connected directory server. For more information, see "Configuring a Customized External Authentication Plug-in" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

Oracle Unified Directory and Oracle Directory Server Enterprise Edition back-end directories uses pass-through authentication for passing authentication through to a connected directory like Microsoft Active Directory for users coming from Oracle Unified Directory or Oracle Directory Server Enterprise Edition. For more information, see:

- The section "Understanding Pass-Through Authentication" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Unified Directory*.

- The section "[Pass-Through Authentication](#)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.

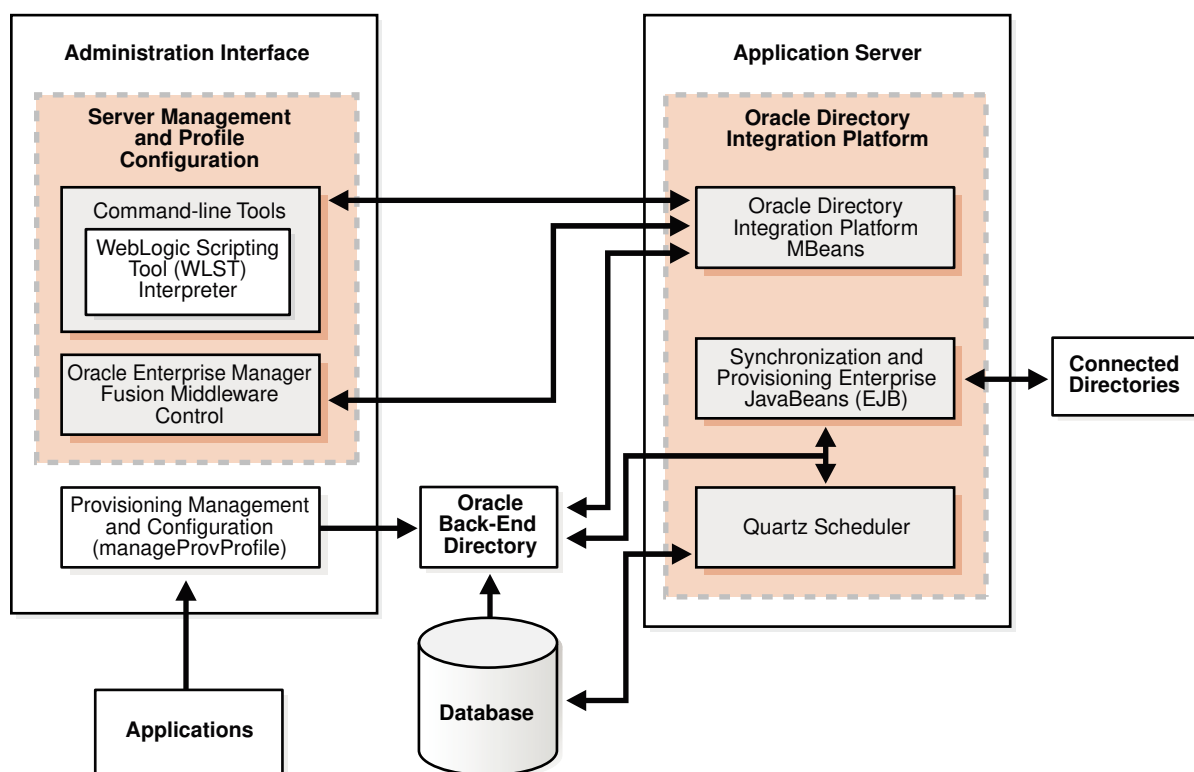
## 1.4.2 About Oracle Directory Integration Platform

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.

 **Note:**

Figure 1-1, shows an external database for Oracle Internet Directory. Oracle Unified Directory and Oracle Directory Server Enterprise Edition uses an internal database.

### 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:**

[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:

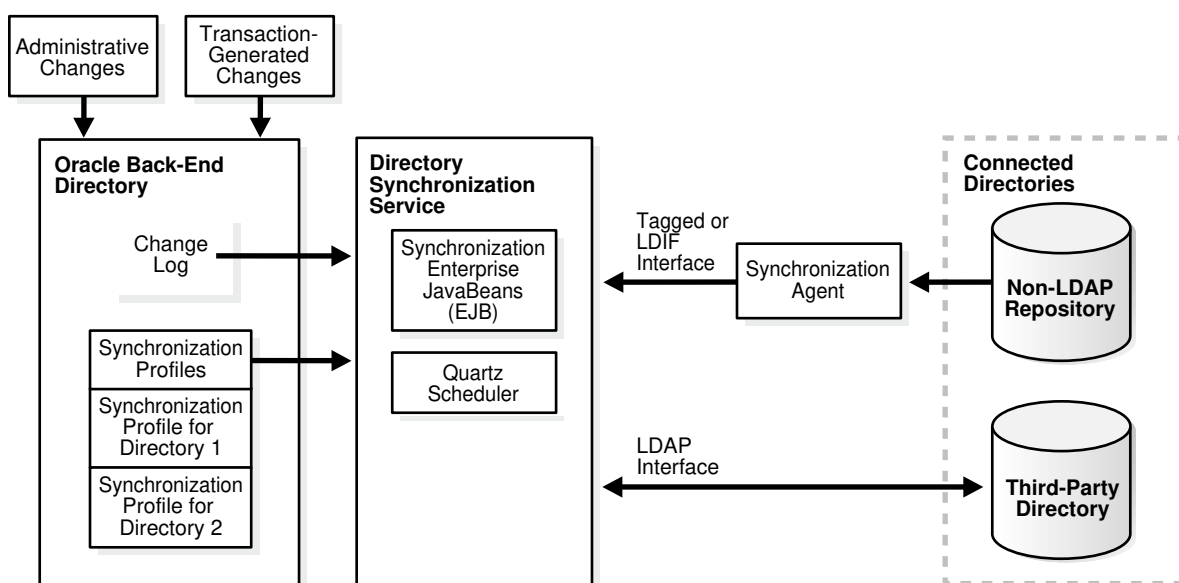
- One-way: Some connected directories only supply changes to the Oracle back-end directory and do not receive changes from it. This is the case, for example, with Oracle Human Resources, the primary repository and basis for comparison for employee information.

- 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, 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.

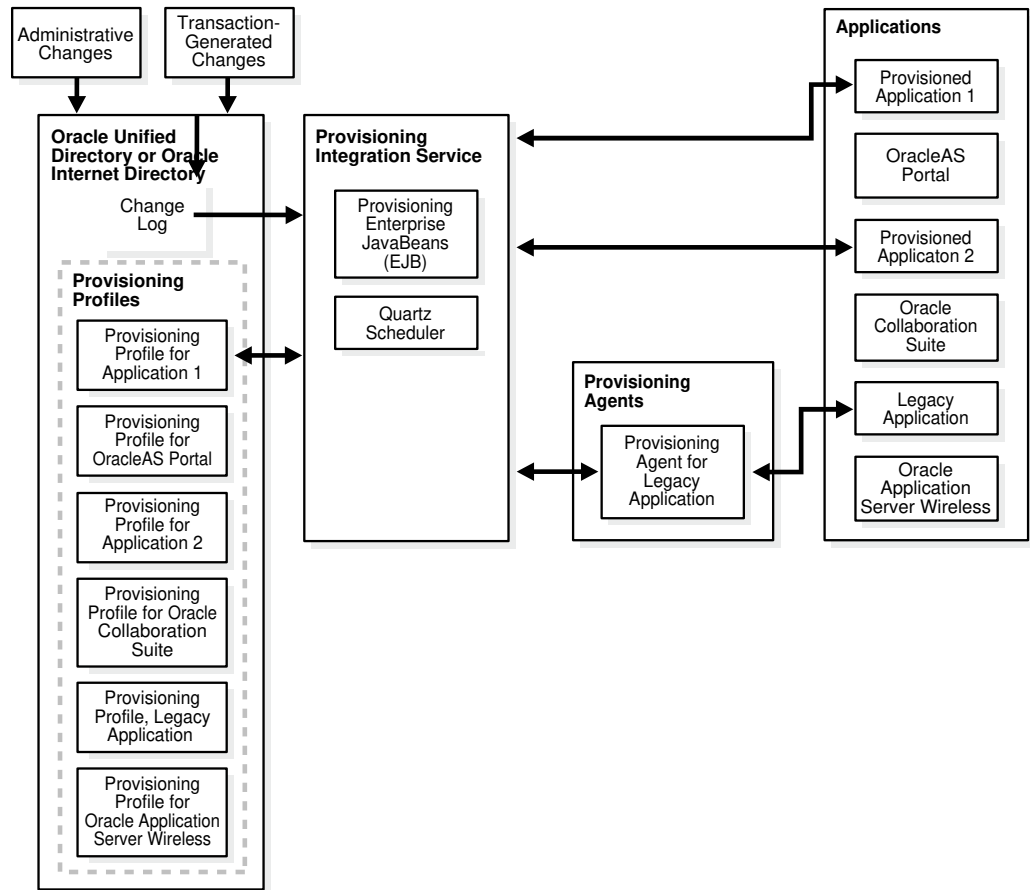
 **Note:**

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**



# 2

## Security Features in Oracle Directory Integration Platform

This chapter discusses the most important aspects of security in Oracle Directory Integration Platform.

**Topics:**

- [Overview of Authentication in Oracle Directory Integration Platform](#)
- [About Access Control and Authorization and Oracle Directory Integration Platform](#)
- [About Data Integrity and Oracle Directory Integration Platform](#)
- [About Data Privacy and Oracle Directory Integration Platform](#)
- [About Tools Security and Oracle Directory Integration Platform](#)
- [Credential Store Framework for Oracle Directory Integration Platform](#)

### 2.1 Overview of 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.

**Topics:**

- [Secure Sockets Layer and Oracle Directory Integration Platform](#)
- [Oracle Directory Integration Platform Authentication in SSL Mode](#)
- [Transparent Layer Security Protocol and Cipher Suites](#)
- [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 [Secure Socket Layer \(SSL\)](#) 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:**

If the Oracle back-end directory is Oracle Internet Directory, then Oracle Directory Integration Platform supports both No Authentication SSL mode (SSL mode 1) and SSL Server Authentication (SSL mode 2). If Oracle Unified Directory or Oracle Directory Server Enterprise Edition is your Oracle back-end directory, SSL Server Authentication (SSL mode 2) is the only SSL option.

Oracle does not recommend using No Authentication (SSL Mode 1).

- SSL Server Authentication (SSL Mode 2)—Oracle recommends this mode for Oracle Directory Integration Platform. It 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.

 **Note:**

Oracle Directory Integration Platform 12c supports Transport Layer Security (TLS) v1.2 protocol for communication with connected directories. See [Transport Layer Security Protocol and Cipher Suites](#).

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 2, then Directory Integration Platform must be configured to connect to the Oracle back-end directory using the same SSL mode 2.

 **See Also:**

If using Oracle Internet Directory as the Oracle back-end directory, refer to the *Oracle Fusion Middleware Administering 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](#)".

## 2.1.3 Transport Layer Security Protocol and Cipher Suites

Oracle Directory Integration Platform 12c supports Transport Layer Security (TLS) v1.2 protocol for communication with connected directories and uses the cipher suites to encrypt and decrypt data.

The TLS protocol is a successor to the SSL protocol. TLS is a widely used protocol today by applications that entails data transmission over a network. The primary goal of the TLS protocol is to provide enhanced security and data integrity between two communicating applications.

The Transport Layer Security protocol now has three versions for increasing security: TLS 1.0, TLS 1.1, and TLS 1.2. The default version used in Oracle Directory Integration Platform is TLS 1.2.

### Topics:

- [Supported Out-of-Box Cipher Suites](#)
- [Disabled Cipher Suites in Oracle Directory Integration Platform](#)
- [Adding Cipher Suites into Oracle Directory Integration Platform](#)
- [Specify the SSL/TLS Protocol Version](#)

### 2.1.3.1 Supported Out-of-Box Cipher Suites

Oracle Directory Integration Platform supports a large number of cipher suites which is available out-of-box.

During a TLS handshake, the Oracle Directory Integration Platform and connected directory initiates the connection using TLSv1.2 protocol and supported ciphers in the order listed below.

#### Note:

Oracle Directory Integration Platform provides you the option to add cipher suites used by the connected directories, which is not available out-of-box. See [Adding Cipher Suites into Oracle Directory Integration Platform](#).

**Table 2-1 Out-of-Box Cipher Suites**

TLS Protocol	Cipher Suite
TLS version 1.2	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS version 1.2	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS version 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

**Table 2-1 (Cont.) Out-of-Box Cipher Suites**

TLS Protocol	Cipher Suite
TLS version 1.2	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS version 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS version 1.2	TLS_RSA_WITH_AES_128_GCM_SHA256
TLS version 1.2	TLS_RSA_WITH_AES_256_GCM_SHA384
TLS version 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS version 1.2	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS version 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS version 1.2	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS version 1.2	TLS_DHE_DSS_WITH_AES_128_GCM_SHA256
TLS version 1.2	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS version 1.2	TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
TLS version 1.2	TLS_DHE_DSS_WITH_AES_256_GCM_SHA384
TLS version 1.2	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS version 1.2	TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
TLS version 1.2	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS version 1.2	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
TLS version 1.2	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS version 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS version 1.2	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS version 1.2	TLS_DHE_DSS_WITH_AES_128_CBC_SHA
TLS version 1.2	TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS version 1.2	TLS_DHE_DSS_WITH_AES_256_CBC_SHA
TLS version 1.2	TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS version 1.2	TLS_DH_DSS_WITH_AES_128_GCM_SHA256
TLS version 1.2	TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
TLS version 1.2	TLS_DH_DSS_WITH_AES_256_GCM_SHA384
TLS version 1.2	TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
TLS version 1.2	TLS_RSA_WITH_AES_128_CBC_SHA256
TLS version 1.2	TLS_DH_DSS_WITH_AES_128_CBC_SHA256
TLS version 1.2	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
TLS version 1.2	TLS_RSA_WITH_AES_256_CBC_SHA256
TLS version 1.2	TLS_DH_DSS_WITH_AES_256_CBC_SHA256
TLS version 1.2	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
TLS version 1.2	TLS_RSA_WITH_AES_128_CBC_SHA
TLS version 1.2	TLS_DH_DSS_WITH_AES_128_CBC_SHA
TLS version 1.2	TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
TLS version 1.2	TLS_RSA_WITH_AES_256_CBC_SHA
TLS version 1.2	TLS_DH_DSS_WITH_AES_256_CBC_SHA
TLS version 1.2	TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA
TLS version 1.2	TLS_DH_RSA_WITH_AES_128_GCM_SHA256
TLS version 1.2	TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
TLS version 1.2	TLS_DH_RSA_WITH_AES_256_GCM_SHA384
TLS version 1.2	TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384

**Table 2-1 (Cont.) Out-of-Box Cipher Suites**

TLS Protocol	Cipher Suite
TLS version 1.2	TLS_DH_RSA_WITH_AES_128_CBC_SHA256
TLS version 1.2	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
TLS version 1.2	TLS_DH_RSA_WITH_AES_256_CBC_SHA256
TLS version 1.2	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
TLS version 1.2	TLS_DH_RSA_WITH_AES_128_CBC_SHA
TLS version 1.2	TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
TLS version 1.2	TLS_DH_RSA_WITH_AES_256_CBC_SHA
TLS version 1.2	TLS_ECDH_RSA_WITH_AES_256_CBC_SHA
TLS version 1.2	TLS_RSA_WITH_CAMELLIA_256_CBC_SHA
TLS version 1.2	TLS_RSA_WITH_CAMELLIA_128_CBC_SHA
TLS version 1.2	TLS_RSA_WITH_3DES_EDE_CBC_SHA

### 2.1.3.2 Disabled Cipher Suites in Oracle Directory Integration Platform

The following is a list of weak cipher suites that are disabled by default in Oracle Directory Integration Platform:

- SSL\_RSA\_WITH\_RC4\_128\_MD5
- SSL\_RSA\_WITH\_RC4\_128\_SHA
- TLS\_ECDHE\_ECDSA\_WITH\_RC4\_128\_SHA
- TLS\_ECDHE\_RSA\_WITH\_RC4\_128\_SHA
- TLS\_ECDH\_ECDSA\_WITH\_RC4\_128\_SHA
- TLS\_ECDH\_RSA\_WITH\_RC4\_128\_SHA
- SSL\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
- SSL\_DHE\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
- SSL\_DHE\_DSS\_WITH\_3DES\_EDE\_CBC\_SHA
- TLS\_EMPTY\_RENEGOTIATION\_INFO\_SCSV

### 2.1.3.3 Adding Cipher Suites into Oracle Directory Integration Platform

If the cipher suites configured in the back-end or connected directory are not available or recognized in Oracle Directory Integration Platform then you must add those suites into Oracle Directory Integration Platform using the Oracle Fusion Middleware System MBean Browser.

To add cipher suites into Oracle Directory Integration Platform, complete the following steps:

 **Note:**

- In a cluster environment, you must repeat the below steps for all of the Oracle Directory Integration Platform managed servers in the cluster.
- You can also use the WLST command to add the cipher suites. See *Setting Cipher Suites Using WLST: An Example in Oracle Fusion Middleware Administering Security for Oracle WebLogic Server*.

1. Open a browser, and access the Fusion Middleware Control Console using the following URL format:

`http://host1.example.com:7001/em`

2. Enter the Oracle Fusion Middleware administrator user name and password and click **Login**.
3. From the target navigation pane, expand the domain.
4. From the domain home page, select the Managed Server (`wls_ods1`).

 **Note:**

The default value for Oracle Directory Integration Platform Managed Server is `wls_ods1`.

5. From the WebLogic Server menu, choose System **MBean Browser**. The **System MBean Browser** page is displayed.

 **Note:**

You can also click the **Find** icon to perform a search for an MBean, attribute. For example `com.bea:Name=wls_ods1,Type=Server`.

6. Expand **Configuration MBeans** in the MBean navigation tree and then select **com.bea > Server**.
7. Expand the Server node and then expand the Managed Server node (`wls_ods1`).
8. From the Managed Server node, expand SSL and then select the Managed Server MBEAN.  
The **Configuration MBEAN** page is displayed.
9. Select **Attributes** tab and then select **CipherSuites**.  
The **Attribute: Ciphersuites** page is displayed.
10. Click **Add** and then add the ciphers configured in the back-end or connected directory.
11. Click **Apply**.

 **Note:**

After you add the ciphers, the out-of-box supported cipher suites in Oracle Directory Integration Platform are not recognized. You must add the required default ciphers for communication between the Oracle Directory Integration Platform and back-end or connected directories. See [Supported Out-of-Box Cipher Suites](#).

### 2.1.3.4 Specify the SSL/TLS Protocol Version

You can use the Fusion Middleware Control Console to specify the SSL/TLS protocol versions enabled for the SSL handshake. While in most cases the most recent version of the SSL or TLS protocol is desirable, peers may not support it. You may want to specify the enabled SSL or TLS protocol based on circumstances (compatibility, SSL performance, and environments with maximum security requirements) .

To specify TLS protocol version in Oracle Directory Integration Platform, complete the following steps:

1. Open a browser, and access the Fusion Middleware Control Console using the following URL format:

```
http://host1.example.com:7001/em
```

2. Enter the Oracle Fusion Middleware administrator user name and password and click **Login**.
3. From the target navigation pane, expand the domain.
4. From the domain home page, select the Managed Server (`wls_ods1`).

 **Note:**

The default value for Oracle Directory Integration Platform Managed Server is `wls_ods1`.

5. From the WebLogic Server menu, choose System **MBean Browser**. The **System MBean Browser** page is displayed.

 **Note:**

You can also click the **Find** icon to perform a search for an MBean, attribute. For example `com.bea:Name=wls_ods1,Type=Server`.

6. Expand **Configuration MBeans** in the MBean navigation tree and then select **com.bea > Server**.
7. Expand the Server node and then expand the Managed Server node (`wls_ods1`).
8. From the Managed Server node, expand SSL and then select the Managed Server MBEAN.  
The **Configuration MBEAN** page is displayed.
9. Select **Attributes** tab and then select **MinimumTLSProtocolVersion**.



The **Attribute: MinimumTLSProtocolVersion** page is displayed.

10. You can specify the following versions in the **Value** field:

- `SSLv3`: Specifies SSL V3.0 as the minimum protocol version enabled in SSL connections.
- `TLSv1`: Specifies TLS V1.0 as the minimum protocol version enabled in SSL connections.
- `TLSvx.y`: Specifies TLS V<sub>x.y</sub> as the minimum protocol version enabled in SSL connections, where *x* is an integer between 1 and 9 (inclusive) and *y* is an integer between 0 and 9 (inclusive). For example, `TLSv1.2`.

 **Note:**

The protocol that you specify here is the lowest supported version of SSL and TLS that are enabled for SSL handshake. It does not guarantee that the specified version will be picked up. The version in effect is determined by the underlying JRE which protocol that is enabled and also the SSL handshake between Oracle Directory Integration Platform and connected/back-end directories.

For example, if `SSLV3` is specified then the `SSLV3`, `TLSv1.0`, `TLSv1.1` and `TLSv1.2` protocols are supported. It depends on the underlying JRE & SSL handshake and the required protocol would be picked for SSL communication.

11. Click **Apply**.

 **Note:**

In a cluster environment, you must repeat the above steps for all of the Oracle Directory Integration Platform managed servers in the cluster.

## 2.1.4 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 back-end directory the administrator is represented by the DN `cn=dipadmin,cn=dipadmins,cn=directory integration platform,cn=products,cn=oraclecontext`.
- Members of the Oracle Directory Integration Platform administrator group.  
In back-end directory the administrator group is represented by the DN `cn=odisgroup,cn=dipadmins,cn=directory integration platform,cn=products,cn=oraclecontext`.

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 About 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.

- If the Oracle Unified Directory is used as the back-end directory, then some privileges are assigned to the Oracle Directory Integration Platform user. For more information, see [Understanding Root Users and the Privilege Subsystem in Administering Oracle Unified Directory](#).
- If the Oracle Internet Directory is used as the back-end directory, then 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.

The following section discusses these policies:

- [Understanding Access Controls for the Oracle Directory Integration Platform](#)
- [Understanding Access Controls for Profiles](#)

### 2.2.1 Understanding 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.

- 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 contacting Oracle Unified Directory, 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.
- When contacting Oracle Unified Directory or Oracle Directory Server Enterprise Edition, the directory integration server binds using the Oracle Directory Integration Platform user and then uses the proxy authorization v2 control (RFC 4370) to perform the actions against the directory server acting as 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 back-end directory the DN of `odisgroup` is:

```
cn=odisgroup,cn=directory_admins,cn=directory_integration  
platform,cn=products,cn=oraclecontext
```

You control the access rights granted to directory integration servers by placing access control policies for 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 Understanding 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 `odipgroup` and `odipegroup` groups are also created during configuration using the `dipConfigurator` command. All import profiles are assigned to the `odipgroup` group and all export profiles are assigned to the `odipegroup` group. Rights are controlled by placing appropriate access policies for 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 back-end directory change logs, to which access is otherwise restricted.

### See Also:

- The chapter "[Controlling Access To Data](#)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Unified Directory*.
- The chapter "[Managing Directory Access Control](#)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.
- The chapter "[Directory Server Access Control](#)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.

## 2.3 About 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 About 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 directory, you must run both components in the same SSL mode.

## 2.5 About Tools Security and Oracle Directory Integration Platform

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

## 2.6 Credential Store Framework for Oracle Directory Integration Platform

Oracle Directory Integration Platform uses the Oracle Fusion Middleware infrastructure Credential Store Framework. Know more about the credentials Oracle Directory Integration Platform stores in this Credential Store Framework.

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 name:
 

```
cn=odisrv,cn=Registered Instances,cn=Directory Integration Platform,cn=products,cn=oraclecontext
```
- 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 Administration Server using the `connect()` command, enter:

```
createCred(map="dip", key="jksKey", user="jksUser", password="password", desc="DIP SSL JKS")
```

**Table 2-2 createCred Commands**

Argument	Description
map	Specifies the map name (folder) of the credential. The map option is fixed and the only supported value is dip.
key	Specifies the key name of the credential. The key option is fixed and the only supported values is jksKey.
user	Specifies the credential user name. The user attribute is not used to access the keystore and the JKS password attribute is used to access the keystore. You can specify any value for the user.
password	Specifies the credential password. The password attribute is used to access the keystore.
desc	Specifies a string describing the credential.

You can change the keystore password to a new value, by running the `updateCred()` command:

```
updateCred(map="dip", key="jksKey", user="jksUser", password="password")
```

 **Note:**

- See Introduction to Oracle Fusion Middleware Audit Framework in *Oracle Fusion Middleware Securing Applications with Oracle Platform Security Services*.
- 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

Know more about 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:

- [Administering Oracle Directory Integration Platform](#)
- [Managing the Oracle Directory Integration Platform](#)

# 3

## Administering Oracle Directory Integration Platform

This chapter describes tools you can use to administer Oracle Directory Integration Platform.

**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

After you install and configure Oracle Directory Integration Platform, you can use the graphical tools to administer your environment.

Oracle offers the following graphical tools for Oracle Directory Integration Platform:

- [Using Fusion Middleware Control](#)
- [Using Oracle Unified Directory Services Manager for Oracle Unified Directory](#)
- [Using Oracle Directory Services Manager for Oracle Internet Directory](#)
- [Using Directory Service Control Center for Oracle Directory Server Enterprise Edition](#)

 **Note:**

Prior to 11g 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 12c you must use Oracle Enterprise Manager Fusion Middleware Control.

#### 3.1.1 Using Fusion Middleware Control

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. Open a browser, and access the Fusion Middleware Control Console using the following URL format:  

```
https://host:port/em
```
2. Enter the Oracle Fusion Middleware administrator user name and password and click **Login**.
3. From the target navigation pane, expand the domain.
4. From the domain home page, select the Managed Server (wls\_ods1).

 **Note:**

The default value for Oracle Directory Integration Platform Managed Server is wls\_ods1.

5. Use the WebLogic Server menu, to select tasks.

### 3.1.1.1 About 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.
- 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 Unified Directory Services Manager for Oracle Unified Directory

Oracle Unified Directory Services Manager (OUDSM) is a web-based interface for managing instances of Oracle Unified Directory. OUDSM enables you to configure the structure of the directory, define objects in the directory, add and configure users, groups, and other directory features.

See *Accessing Oracle Unified Directory Using OUDSM* in *Oracle Fusion Middleware Administering Oracle Unified Directory*.

### 3.1.3 Using Oracle Directory Services Manager for Oracle Internet Directory

Oracle Directory Services Manager is a web-based interface for managing Oracle Internet Directory instances. Oracle Directory Services Manager enables you to configure the structure of the directory, define objects in the directory, add and configure users, groups, and other entries.

See *Overview of Oracle Directory Services Manager* in *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.



 **Note:**

For Oracle Internet Directory, Oracle Directory Services Manager is a replacement for Oracle Directory Manager, which is now deprecated.

### 3.1.4 Using Directory Service Control Center for Oracle Directory Server Enterprise Edition

Directory Service Control Center (DSCC) is a console that enables you to manage Oracle Directory Server Enterprise Edition using a web browser.

See [Using the Directory Service Control Center \(DSCC\)](#) in *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.

## 3.2 Command-Line Tools for Administering Oracle Directory Integration Platform

You can also use the command-line tools for administering Oracle Directory Integration Platform.

The following command-line tools, located in the `ORACLE_HOME/bin` directory:

 **Note:**

- 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 with the back-end directory. For more information, see "[Viewing the Status of Oracle Directory Integration Platform Using the dipStatus Utility](#)".
  - `dipConfigurator`: Allows you to configure Oracle Directory Integration Platform with Oracle Unified Directory, Oracle Internet Directory, and Oracle Directory Server Enterprise Edition. For more information see:
    - [Configuring Oracle Directory Integration Platform for Oracle Unified Directory](#)
    - [Configuring Oracle Directory Integration Platform for Oracle Internet Directory](#)
    - [Configuring Oracle Directory Integration Platform for Oracle Directory Server Enterprise Edition](#)

- `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 "[Manage Oracle Directory Integration Platform Using `manageDIPServerConfig`](#)" for more information.
- `manageSyncProfiles`: Manages Oracle Directory Integration Platform synchronization profiles. Refer to "[Managing Synchronization Profiles Using `manageSyncProfiles`](#)" 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 "[Overview of Directory Bootstrapping Using `syncProfileBootstrap`](#)" 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`](#)" 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](#)" for more information.
- `manageProvProfiles`: 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 "[About `manageProvProfiles` Command](#)" for more information.

# 4

## Managing the Oracle Directory Integration Platform

This chapter discusses the Oracle Directory Integration Platform and explains how to configure and manage it.

### Topics:

- [Understanding 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](#)
- [Manage Oracle Directory Integration Platform Using `manageDIPServerConfig`](#)
- [About Oracle Unified Directory Configuration for SSL Mode](#)
- [About Oracle Directory Server Enterprise Edition Configuration for SSL Mode](#)
- [About Oracle Internet Directory Configuration for SSL Mode](#)
- [Managing the SSL Certificates of Back-End Directories and Connected Directories](#)
- [About Oracle Directory Integration Platform in a High Availability Scenario](#)
- [Understanding How to Manage Oracle Directory Integration Platform in a Replicated Environment](#)

### See Also:

"[About Oracle Directory Integration Platform](#)" for a summary of the functions performed by the Oracle Directory Integration Platform

### Note:

For synchronizing the password, Oracle recommends using the SSL communication to connect the Oracle Directory Integration Platform to the back-end directory and any connected directory.

## 4.1 Understanding Operational Information About the Oracle Directory Integration Platform

Know more about the Oracle Directory Integration Platform structural and operational information.

**Topics:**

- [What are Directory Integration Profiles?](#)
- [Understanding Oracle Directory Integration Platform Event Propagation in a Multimaster Oracle Back-end Directory Replication Environment](#)

### 4.1.1 What are 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 Understanding 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 Synchronizing Directory 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 About Directory Provisioning in an Oracle Unified Directory or Oracle Internet Directory Multimaster Replication Environment

In a default multimaster Oracle Unified Directory or Oracle Internet Directory replication environment, the Oracle Directory Integration Platform is installed in the same location as the primary Oracle Unified Directory or 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 Unified Directory or 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.

 **Note:**

The "Oracle Directory Integration Platform Tools" chapter in *Oracle Fusion Middleware Reference for Oracle Identity Management* for more information on the `manageSyncProfiles` command.

## 4.2 Viewing Oracle Directory Integration Platform Status and Registration Information

You can use the `dipStatus` utility and `ldapsearch` utility to view the Oracle Directory Integration Platform status and registration information.

**Topics:**

- [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

Run the `dipStatus` utility to view the status of Oracle Directory Integration Platform.

For more information, see [dipStatus Utility](#).

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

You can view registration information for the Oracle Directory Integration Platform component by running the `ldapsearch` utility and perform a base search on its entry.

**Example:**

```
ldapsearch -h backend_host -p backend_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}kEQc1+/AZEXVukeA5YPnog==
```

## 4.3 Managing Oracle Directory Integration Platform Using Fusion Middleware Control

You can use Oracle Enterprise Manager Fusion Middleware Control to manage Oracle Directory Integration Platform.

**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

You can view runtime information for the Oracle Directory Integration Platform component using Oracle Enterprise Manager Fusion Middleware Control.

To do so, perform the following steps:

1. Open a browser, and access the Fusion Middleware Control Console using the following URL format:

```
http://host1.example.com:7001/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.

 **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

You can start Oracle Directory Integration Platform by using Oracle Enterprise Manager Fusion Middleware Control.

Perform the following:

1. Open a browser, and access the Fusion Middleware Control Console using the following URL format:  
`http://host1.example.com:7001/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, select **Control**, and then click **Start Up**.

## 4.3.3 Stopping Oracle Directory Integration Platform with Fusion Middleware Control

You can stop Oracle Directory Integration Platform by using Oracle Enterprise Manager Fusion Middleware Control.

To do so, perform the following steps:

1. Open a browser, and access the Fusion Middleware Control Console using the following URL format:  
`http://host1.example.com:7001/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, select **Control**, and then click **Shut Down**.
5. When the confirmation dialog appears, click **Yes**.



## 4.3.4 Managing the Oracle Directory Integration Platform Server Configuration

You can 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

To do so, perform the following:

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-SSL (mode 0): Directory Integration Platform connects to the Oracle back-end directory without SSL encryption.

- 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.

 **Note:**

Oracle does not recommend using No Authentication (SSL Mode 1).

- 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 "[About Oracle Internet Directory Configuration for SSL Mode](#)" 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.

 **Note:**

See Managing Log Files and Diagnostic Data in *Oracle Fusion Middleware Administering Oracle Fusion Middleware* 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 Oracle Fusion Middleware Audit Framework of the Oracle Platform Security Services for compliance, monitoring, and analytic purposes. Using Oracle Enterprise Manager Fusion Middleware Control, you can view, search, and manage audit data and event settings for Oracle Directory Integration Platform.

See Introduction to Oracle Fusion Middleware Audit Framework in *Oracle Fusion Middleware Securing Applications with Oracle Platform Security Services*.

## 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 Manage Oracle Directory Integration Platform Using `manageDIPServerConfig`

The Manage DIP Server Configuration utility, `manageDIPServerConfig`, allows you to manage the Oracle Directory Integration Platform server configuration.

For more information, see [manageDIPServerConfig Utility](#).

 **Note:**

- 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.

## 4.6 About Oracle Unified Directory Configuration for SSL Mode

For more information, see [Configuring Oracle Unified Directory \(SSL\) for Oracle Directory Integration Platform](#).

## 4.7 About Oracle Directory Server Enterprise Edition Configuration for SSL Mode

For more information, see [Configuring Oracle Directory Server Enterprise Edition \(SSL\) for Oracle Directory Integration Platform](#).

## 4.8 About Oracle Internet Directory Configuration for SSL Mode

For more information, see [Configuring Oracle Internet Directory \(SSL\) for Oracle Directory Integration Platform](#).

 **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 *Configuring SSL in Oracle Fusion Middleware* in *Oracle Fusion Middleware Administering Oracle Fusion Middleware* for complete information about the Oracle SSL Automation Tool.

## 4.9 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:

- ["Manage 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.

### 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*.

## 4.10 About Oracle Directory Integration Platform in a High Availability Scenario

For more information, see Oracle Directory Integration Platform High Availability in *Oracle Fusion Middleware High Availability Guide*.

## 4.11 Understanding How to Manage 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.

## 4.12 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.

### Note:

- 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.

### dipStatus Syntax

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

### Arguments for dipStatus

The following table describes the arguments for `dipStatus` utility.

Table 4-1 dipStatus utility Arguments

Argument	Description
<b>-h   -host</b>	Oracle WebLogic Server where Oracle Directory Integration Platform is deployed.
<b>-p   -port</b>	Listening port of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed.
<b>-D   -wlsuser</b>	Oracle WebLogic Server login ID. <b>Note:</b> 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 <code>dipStatus</code> 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.
<b>-ssl</b>	Executes the command in SSL mode. <b>Note:</b> The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. For more information, see "Configuring SSL" in <i>Oracle Fusion Middleware Securing Oracle WebLogic Server</i> .
<b>-keystorePath</b>	The full path to the keystore.
<b>-keystoreType</b>	The type of the keystore identified by <code>-keystorePath</code> . For example: <code>-keystorePath jks</code> or <code>-keystorePath PKCS12</code>
<b>-help</b>	Provides usage help for the command.

### Examples for dipStatus

The following shows an example of `dipStatus` utility command.

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

```
dipStatus -help
```

## 4.13 manageDIPServerConfig Utility

Use the `manageDIPServerConfig` utility to manage the Oracle Directory Integration Platform server configuration. This utility is located in the `ORACLE_HOME/bin` directory.

### manageDIPServerConfig Syntax

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

### Arguments for manageDIPServerConfig

The following table describes the arguments for `manageDIPServerConfig` utility.

Table 4-2 dipStatus utility Arguments

Argument	Description
<b>get   set</b>	Operation to perform. <ul style="list-style-type: none"><li>• <b>get</b>: Displays the current value of the config parameter in DIP configuration file.</li><li>• <b>set</b>: Updates the value of the config parameter in DIP configuration file.</li></ul>
<b>-h   -host</b>	Oracle WebLogic Server where Oracle Directory Integration Platform is deployed.
<b>-p   -port</b>	Listening port of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed.
<b>-D   -wlsuser</b>	Oracle WebLogic Server login ID. <b>Note:</b> 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 <code>dipStatus</code> 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.



Table 4-2 (Cont.) dipStatus utility Arguments

Argument	Description
-attr   -attribute	<p>Identifies the attribute that <code>manageDIPServerConfig</code> performs the operation on. The following is a list and description of the attributes <code>manageDIPServerConfig</code> can perform operations on:</p> <ul style="list-style-type: none"> <li>• <code>sslmode</code>: The SSL mode Oracle Directory Integration Platform uses to connect to the Oracle back-end directory. Supported values are 0, 1 and 2. Use 0 for non-SSL communication, 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).</li> <li>• <code>refreshinterval</code>: The time interval (amount of time in seconds) that controls how often the Oracle Directory Integration Platform server refreshes profile configuration details.</li> <li>• <code>quartzthreadcount</code>: Controls how many profiles can be scheduled in parallel. The default value is 15. If you have more than 15 profiles, increase the <code>quartzthreadcount</code> attribute accordingly.</li> <li>• <code>quartzdbretryinterval</code>: Controls how often Oracle Directory Integration Platform's Quartz scheduler attempts to reconnect to the Oracle back-end directory database.</li> <li>• <code>quartzjobstore</code>: Stores the scheduling information in the Oracle back-end directory database. The default value is <code>ldapjobstore</code>.</li> <li>• <code>backendhostport</code>: Identifies the host and port of the Oracle back-end directory associated with Oracle Directory Integration Platform. Specify values for the <code>backendhostport</code> attribute in the form of <code>host:port</code>.</li> <li>• <code>keystorelocation</code>: 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 <code>keystorelocation</code> attribute, be sure you use the appropriate path separators (that is, / for UNIX and Linux platforms, and \ for Windows platforms).</li> <li>• <code>isclustered</code>: Specifies if the Oracle Directory Integration Platform instance is in a cluster environment. The default value is <code>false</code>.</li> <li>• <code>clustercheckininterval</code>: Specifies the frequency (milliseconds) at which an instance checks for server status (For example, detecting failed instances) with the other instances of the cluster. The default value is 120000 milliseconds.</li> </ul>
-ssl	<p>Executes the command in SSL mode.</p> <p><b>Note:</b> The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. For more information, see "Configuring SSL" in <i>Oracle Fusion Middleware Securing Oracle WebLogic Server</i>.</p>
-keystorePath	The full path to the keystore.

Table 4-2 (Cont.) dipStatus utility Arguments

Argument	Description
<b>-keystoreType</b>	The type of the keystore identified by <code>-keystorePath</code> . For example: <code>-keystorePath jks</code> or <code>-keystorePath PKCS12</code>
<b>-val   -value</b>	The value to set for the attribute This parameter is required with the set operation.
<b>-help</b>	Provides usage help for the command.

### 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 backendhostport -value backend_host: backend_SSL_port
```

# Part III

## Configuring Oracle Back-End Directory

This part discusses how to configure the Oracle back-end directory for Oracle Directory Integration Platform.

Part III includes the following chapters:

- [Configuring Oracle Unified Directory](#)
- [Configuring Oracle Internet Directory](#)
- [Configuring Oracle Directory Server Enterprise Edition](#)

# 5

## Configuring Oracle Unified Directory

You can configure Oracle Unified Directory as the back-end directory for Oracle Directory Integration Platform synchronization or provisioning.

### Topics:

- [Before You Configure Oracle Unified Directory as the Back-End Directory](#)
- [Configuring Oracle Unified Directory \(Non-SSL\) for Oracle Directory Integration Platform](#)
- [Configuring Oracle Unified Directory \(SSL\) for Oracle Directory Integration Platform](#)
- [Verifying Oracle Directory Integration Platform](#)

### 5.1 Before You Configure Oracle Unified Directory as the Back-End Directory

Before you can configuring Oracle Unified Directory as the as the back-end directory, you must configure Oracle Directory Integration Platform.

For more information about configuring Oracle Directory Integration Platform, see Configuring Oracle Directory Integration Platform in *Installing and Configuring Oracle Internet Directory*.

### 5.2 Configuring Oracle Unified Directory (Non-SSL) for Oracle Directory Integration Platform

Use the steps in the following order to configure Oracle Unified Directory (back-end directory) non-SSL communication for Oracle Directory Integration Platform.

- [Installing Oracle Unified Directory](#)
- [Configuring Oracle Unified Directory](#)
- [Creating Oracle Unified Directory Suffixes](#)
- [Enabling External Change Log](#)
- [Configuring the Oracle WebLogic Server Domain for Oracle Directory Integration Platform with Oracle Unified Directory](#)
- [Starting the Servers](#)
- [Configuring Oracle Directory Integration Platform for Oracle Unified Directory](#)
- [Adding Access Control Instructions \(ACIs\) for Oracle Unified Directory](#)

## 5.2.1 Installing Oracle Unified Directory

Install the Oracle Unified Directory either as a collocated configuration or as a standalone configuration.

To install Oracle Unified Directory, see *Installing the Oracle Unified Directory Software in Oracle Fusion Middleware Installing Oracle Unified Directory*.

For OUD Oracle home directory location, Oracle recommends that you specify the Oracle Directory Integration Platform home directory, as the Middleware home.

When you set up an Oracle Unified Directory server instance using either the graphical user interface (GUI) or the command-line interface (CLI), ensure that you select one of the following options:

- **Enable for DIP:** Select this option if you want this server instance to be enabled for Oracle Directory Integration Platform (DIP) only.
- **Enable for EBS (E-Business Suite), Database Net Services and DIP:** Select this option if you want this server instance to be enabled for Oracle E-Business Suite (EBS), Oracle Database Net Services, and Oracle Directory Integration Platform (DIP).
- **Enable for EUS (Enterprise User Security), EBS, Database Net Services and DIP:** Select this option if you want this server instance to be enabled for Oracle Enterprise User Security (EUS), Oracle E-Business Suite (EBS), Oracle Database Net Services, and Oracle Directory Integration Platform (DIP).



### Note:

All the above options are valid for Oracle Directory Integration Platform. Oracle recommends you to use **Enable for DIP** option for integrating Oracle Unified Directory with Oracle Directory Integration Platform and if you are not integrating with EBS, EUS, or Database Net Service.

## 5.2.2 Configuring Oracle Unified Directory

Configure the Oracle Unified Directory, before you use it as the back-end directory for Oracle Directory Integration Platform.

See *Introduction to Oracle Unified Directory in Oracle Fusion Middleware Administering Oracle Unified Directory*.

## 5.2.3 Creating Oracle Unified Directory Suffixes

If you have not created the suffixes during the Oracle Unified Directory installation, then you must create them using the `setup-oracle-context` command.

Create the `cn=oraclecontext` and `cn=oracleschemaversion` suffixes, by running the `setup-oracle-context` command on the command line:

UNIX

```
$ setup-oracle-context -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --no-prompt --trustAll
```

#### Windows

```
setup-oracle-context -h localhost -p 4444 -D "cn=directory manager" -j pwd-file --no-prompt --trustAll
```

## 5.2.4 Enabling External Change Log

The External Change Log (ECL) is available by default on any server instance that includes both a directory server and a replication server.

Enable the ECL for the user suffix and `cn=oraclecontext` using the `dsreplication` command.

### Note:

If you have configured replication during installation then ECL is enabled. For more information, see [Setting Up Replication During Installation](#) in *Installing Oracle Unified Directory*.

#### Enable ECL for the User Suffix

To enable ECL for the user suffix (For example:`dc=example,dc=com`):

#### UNIX

```
$ dsreplication enable-changelog -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -r 8989 -b "dc=example,dc=com" --trustAll --no-prompt
```

#### Windows

```
dsreplication enable-changelog -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -r 8989 -b "dc=example,dc=com" --trustAll --no-prompt
```

#### Enable ECL for the `cn=oraclecontext`

To enable ECL for `cn=oraclecontext`:

#### UNIX

```
$ dsreplication enable-changelog -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -r 8989 -b cn=oraclecontext --trustAll --no-prompt
```

#### Windows

```
dsreplication enable-changelog -h localhost -p 4444 -D "cn=directory manager" -j pwd-file -r 8989 -b cn=oraclecontext --trustAll --no-prompt
```

The replication port (`-r`) is required to configure the ECL, even on a standalone server, because the ECL relies on the replication mechanism. You need only specify the replication port if the change log (or replication) was not previously configured on the server. The default value of the replication port is 8989.

#### Verify ECL for the User Suffix and `cn=oraclecontext`

To verify that the ECL is configured on a directory server instance, run the following search command and look for the `cn=changelog` naming context:

```
$ ldapsearch -h localhost -p 1389 -D "cn=directory manager" -j pwd-file -s base
-b "" "objectclass=*" namingContexts
dn:
namingContexts: cn=changelog
namingcontexts: cn=OracleContext
namingcontexts: cn=OracleSchemaVersion
namingcontexts: dc=example,dc=com
```

## 5.2.5 Configuring the Oracle WebLogic Server Domain for Oracle Directory Integration Platform with Oracle Unified Directory

You must configure Oracle Directory Integration Platform with Oracle Unified Directory either in an existing or in a new WebLogic server Domain.

- [Configuring Oracle Directory Integration Platform with Oracle Unified Directory in an Existing WebLogic Domain](#)
- [Configuring Oracle Directory Integration Platform and Oracle Unified Directory in a New Oracle WebLogic Server Domain](#)

### 5.2.5.1 Configuring Oracle Directory Integration Platform with Oracle Unified Directory in an Existing WebLogic Domain

Perform the following steps to configure Oracle Directory Integration Platform with Oracle Unified Directory an existing WebLogic administration domain:

1. Run the `ORACLE_HOME/oracle_common/common/bin/config.sh` script (UNIX) or `ORACLE_HOME\oracle_common\common\bin\config.cmd` (Windows).  
The **Configuration Type** screen is displayed.
2. Select **Update an existing domain**, and click **Next**.  
The **Templates** screen is displayed.
3. On the **Templates** screen, select **Update Domain Using Product Templates** and then select **Oracle Directory Integration Platform - 12.2.1.3.0[dip]** domain configuration option.

#### Note:

When you select **Oracle Directory Integration Platform - 12.2.1.3.0 [dip]** option, **Oracle Enterprise Manager 12.2.1.3.0 [em]** is automatically selected.

Click **Next**.

The **JDBC Data Sources** screen is displayed.

4. Make changes if required and then click **Next**  
The **JDBC Data Sources Test** screen is displayed.
5. Select the data sources to test, and click **Test Selected Connections**.

Click **Next**.

The **Database Configuration Type** screen is displayed.

6. Make changes if required and then click **Get RCU Configuration** to retrieve the schema information. After successfully retrieving the schema information, click **Next** to continue.

The **JDBC Component Schema** screen is displayed.

7. Verify that the values populated are correct for all schemas and click **Next**.

The **JDBC Component Schema Test** screen is displayed.

8. You can select the component schema to test, and click **Test Selected Connections**. Wait for one or more connection tests to complete. If you do not want to test connections, deselect all data sources.



#### Note:

In order to test connections, the database to which you are trying to connect must be running.

Click **Next**.

The **Advanced Configuration** screen is displayed.

9. Select **Managed Servers, Clusters, and Machines** option. Click **Next**.

The **Managed Servers** screen is displayed.

10. Specify the Managed Server name and click **Next**.

The **Clusters** screen is displayed.

11. Configure Clusters as required and click **Next**.

The **Machines** screen is displayed.

12. Select the **Machine** tab (for Windows) or **Unix Machine** tab. Click on **Add** and specify the machine name. Click **Next**.

13. If you added a machine on the Configure Machines screen, then the **Assign Servers to Machines** screen appears. On the Assign Servers to Machines screen, assign the Administration Server and the Managed server to the specified machine. Click **Next**.

14. On the **Configuration Summary screen**, review the domain configuration, and click **Update** to start extending the domain.

15. Click **Finish**, once the domain is extended.

Your existing Oracle Unified Directory domain is extended to support Oracle Directory Integration Platform.

## 5.2.5.2 Configuring Oracle Directory Integration Platform and Oracle Unified Directory in a New Oracle WebLogic Server Domain

To configure Oracle Directory Integration Platform and Oracle Unified Directory in a new WebLogic domain:



1. Run the `ORACLE_HOME/oracle_common/common/bin/config.sh` script (UNIX) or `ORACLE_HOME\oracle_common\common\bin\config.cmd` (Windows).

The **Configuration Type** screen is displayed.

2. On the **Configuration Type** screen, select **Create a new domain** and enter the full path for the domain or use the **Browse** button to navigate to the directory in which your domains are located. Click **Next**.

The **Templates** screen is displayed.

3. On the **Templates** screen, make sure **Create Domain Using Product Templates** is selected, and then select **Oracle Directory Integration Platform - 12.2.1.3.0[dip]**.

 **Note:**

When you select **Oracle Directory Integration Platform - 12.2.1.3.0 [dip]** option, the following components are automatically selected:

- **Oracle Enterprise Manager 12.2.1.3.0 [em]**
- **Oracle JRF - 12.2.1.3.0 [oracle\_common]**
- **Weblogic Coherence Cluster Extension 12.2.1.3 [wlserver]**

Click **Next**.

Click The **Application Location** screen is displayed.

4. Click **Browse** and specify the full path to the directory in which you want to store the applications that are associated with the domain.

Click **Next**.

The **Administrator Account** screen is displayed.

5. Specify the user name and password for the default WebLogic Administrator account for the domain.

The password must be at least eight characters and must contain at least one number or special character. Confirm the password and click **Next**.

Make a note of these details as you will need them to start or restart the WebLogic domain in the following procedure.

The **Domain Mode and JDK** screen is displayed.

6. Specify the domain mode and Java Development Kit (JDK).
  - a. Select **Production** in the Domain Mode field.

 **Note:**

If you select **Production** mode as the domain, the node manager has a random username and password assigned to it. Use the WebLogic Server Administration Console to reset the password.

- b. Accept **Oracle Hotspot** as a default JDK location.

- c. Click **Next**.

The **Database Configuration Type** screen is displayed.

7. Select RCU Data. This option instructs the Configuration Wizard to connect to the database's Service Table (STB) schema to automatically retrieve schema information for schemas needed to configure the domain.

 **Note:**

Ensure that you have created the database schemas required for Oracle Unified Directory. See *Creating Database Schemas for the Infrastructure Domain Using the Repository Creation Utility in Oracle Fusion Middleware Installing Oracle Unified Directory*

After selecting RCU Data:

- a. Enter the name of the server hosting the database in the **Host Name** field.

 **Note:**

Ensure that you do not specify localhost in the **Host Name** field.

- b. Enter the database DBMS name, or service name if you selected a service type driver in the **DBMS/Service** field.
- c. Enter the port number on which the database listens.
- d. Enter the username and password for connecting to the database's Service Table schema.
- e. Click Get RCU Configuration to retrieve the schema information. After successfully retrieving the schema information, click **Next** to continue.

The **JDBC Component Schema** screen is displayed.

8. Verify that the values populated are correct for all schemas, and Click **Next**.

The **JDBC Component Schema Test** screen is displayed.

9. Test datasource connections that you just configured.

A green check mark in the Status column indicates a successful test. If you encounter issues, see the error message in the Connection Result Log section of the screen, fix the problem, then test the connection again.

The **Advanced Configuration** screen is displayed.

10. To complete domain configuration, select any of these options:
  - **Administration Server:** Required to properly configure the Administration Server's listen address.
  - **Node Manager:** Required to configure Node Manager.
  - **Topology:** Required to configure the Managed Servers and cluster, and for configuring the machine and targeting Managed Servers to the machine.
  - **Deployments and Services:** Required to target to servers or clusters.

Click **Next**.

11. Review each item on the **Configuration Summary** screen and verify that the information is correct.

To make any changes, go back to a screen by clicking the Back button or selecting the screen in the navigation pane. Domain creation does not start until you click **Create**.

A new WebLogic domain (for example: *base\_domain*) is created to support Oracle Directory Integration Platform and Fusion Middleware Control in the <ORACLE\_HOME>\user\_projects\domains directory (on Windows). On UNIX, the domain is created in the <ORACLE\_HOME>/user\_projects/domains directory.

## 5.2.6 Starting the Servers

After the WebLogic domain configuration is complete, you can start the servers to manage the domain.

Perform the following tasks:

1. Start the Administration Server, Node Manager and Managed Server as described in [Starting and Stopping the Oracle Stack](#).
2. Start the Oracle Unified Directory using the `start-ds` command:

UNIX: `$ start-ds`

Windows: `C:\> start-ds`

## 5.2.7 Configuring Oracle Directory Integration Platform for Oracle Unified Directory

After configuring the Oracle WebLogic Server domain, you must configure Oracle Directory Integration Platform for Oracle Unified Directory.

Complete the following steps:

1. Export the certificate for the Oracle Unified Directory Administration Server instance, by running the following command:

UNIX

```
$ keytool -export-cert -alias admin-cert -keystore config/admin-keystore -storepass:file config/admin-keystore.pin -file oud-server-admin-cert.cer
```

Windows

```
keytool -export-cert -alias admin-cert -keystore config\admin-keystore -storepass:file config\admin-keystore.pin -file oud-server-admin-cert.cer
```

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 admin-cert -file /home/Middleware/asinst_1/OUd/admin/oud-server-admin-cert.cer -keystore /home/Middleware/dip.jks
```

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

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

```
manageDIPServerConfig set -attribute keystorelocation -val
full_path_to_keystore -h weblogic_host -p weblogic_managed_server_port -D
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:

```
$ORACLE_HOME/bin/manageDIPServerConfig set -h localhost -p 7005 -D
wlsuser -attribute keystorelocation -val /home/Middleware/dip.jks
```

The system will prompt for the WebLogic password.

4. Update the Oracle Directory Integration Platform SSL configuration, by running the following command:

UNIX

```
$ORACLE_HOME/bin/manageDIPServerConfig set -attribute sslmode -val 0 -h
localhost -p 7005 -D "weblogic"
```

Windows

```
ORACLE_HOME\bin\manageDIPServerConfig set -attribute sslmode -val 0 -h
localhost -p 7005 -D "weblogic"
```

For more information, see [Arguments for manageDIPServerConfig](#).

5. 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/oracle_common/common/bin/wlst.sh (UNIX) or
ORACLE_HOME\oracle_common\common\bin\wlst.cmd (Windows)
```

- 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")
```

6. Set the WL\_HOME and ORACLE\_HOME environment variables for Oracle Directory Integration Platform.

- Run the `dipConfigurator setup (<ORACLE_HOME>/bin)` command on the command line and enter the following arguments:

 **Note:**

If you are running the `dipConfigurator setup` for a secured LDAP (`isldapssl=true`) then you must configure Oracle Directory Integration Platform for Oracle Unified Directory using SSL communication mode. See [Configuring Oracle Directory Integration Platform for Oracle Unified Directory SSL](#).

**Table 5-1 dipConfigurator Properties for Oracle Unified Directory**

Properties	Description
<code>wlshost</code>	Oracle WebLogic Server host name where Oracle Directory Integration Platform is deployed. The default host name is <code>localhost</code> .
<code>wlsport</code>	Listening port number of the Oracle WebLogic Administration Server where Oracle Directory Integration Platform is deployed. The default port number is <code>7001</code> .
<code>wlsuser</code>	Oracle WebLogic Server login user name.
<code>ldaphost</code>	Oracle Unified Directory host name. The default host name is <code>localhost</code> .
<code>ldapport</code>	Oracle Unified Directory server port number. The default value is <code>636</code> .
<code>isldapssl</code>	Specify <code>true</code> or <code>false</code> , to enable or disable SSL. The default value is <code>true</code> .
<code>ldapuser</code>	The bind DN to connect to the directory.
<code>ldapadminport</code>	The administration port number of the Oracle Unified Directory to which you want to connect. The default port number is <code>4444</code> .
<code>isclustered &lt;BOOLEAN&gt;</code>	Specify if the Oracle Directory Integration Platform instance is in a cluster environment. The default value is <code>false</code> .
<code>clustercheckininterval &lt;INT&gt;</code>	Specify the frequency (milliseconds) at which an instance checks for server status (For example, detecting failed instances) with the other instances of the cluster. The default value is <code>120000</code> milliseconds.

**Example:**

**UNIX**

```
$ORACLE_HOME/bin/dipConfigurator setup -wlshost localhost -wlsport 7001 -wlsuser weblogic -ldaphost oudhost -ldapport 1389 -ldapuser "cn=Directory Manager" -isldapssl false -ldapadminport 4444
```

**Windows**

```
ORACLE_HOME/bin/dipConfigurator setup -wlshost localhost -wlsport 7001 -wlsuser weblogic -ldaphost oudhost -ldapport 1389 -ldapuser "cn=Directory Manager" -isldapssl false -ldapadminport 4444
```

**Note:**

You can view the `dipConfig.log` file, located at `<ORACLE_HOME>/ldap/log/`.

## 5.2.8 Adding Access Control Instructions (ACIs) for Oracle Unified Directory

Add the ACIs in an LDIF file for Oracle Unified Directory using the `ldapmodify` command.

Run the `ldapmodify` command on the command line:

```
ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -w <password> <<EOF
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (target="ldap:///dc=example,dc=com")(version 3.0; acl
"Entry-level DIP permissions"; allow (all,proxy)
groupdn="ldap:///cn=dipadmingrp,cn=DIPadmins,cn=Directory Integration
Platform,cn=Products,cn=oraclecontext"; allow (all,proxy)
groupdn="ldap:///cn=odipigroup,cn=DIPadmins,cn=Directory Integration
Platform,cn=Products,cn=oraclecontext"; )
-
add: aci
aci: (targetattr="*")(version 3.0; acl "Attribute-level DIP permissions";
allow (all,proxy) groupdn="ldap:///cn=dipadmingrp,cn=DIPadmins,cn=Directory
Integration Platform,cn=Products,cn=oraclecontext"; allow (all,proxy)
groupdn="ldap:///cn=odipigroup,cn=DIPadmins,cn=Directory Integration
Platform,cn=Products,cn=oraclecontext"; )
EOF
```

**Note:**

This is an example, and you need to replace the `dc=example,dc=com` ACI with your profile configuration.

## 5.3 Configuring Oracle Unified Directory (SSL) for Oracle Directory Integration Platform

Use the steps in the following order to configure Oracle Unified Directory (back-end directory) SSL communication for Oracle Directory Integration Platform.

- [Configuring Oracle Unified Directory for SSL](#)
- [Configuring Oracle Directory Integration Platform for Oracle Unified Directory SSL](#)

## 5.3.1 Configuring Oracle Unified Directory for SSL

You can configure Oracle Unified Directory (back-end directory) SSL communication for Oracle Directory Integration Platform by completing the following steps:

1. Configure Oracle Unified Directory, as described in [Configuring Oracle Unified Directory \(Non-SSL\) for Oracle Directory Integration Platform](#).
2. Configure Oracle Unified Directory to accept SSL-based connections using a self-signed certificate, as described in "Getting SSL Up and Running Quickly" in the *Oracle Unified Directory*.
3. Export the private key for the Oracle Unified Directory instance, by running the following command:

UNIX

```
$ keytool -exportcert -alias server-cert -file config/server-cert.txt -rfc \
  -keystore config/keystore -storetype JKS
```

Windows

```
keytool -exportcert -alias server-cert -file config/server-cert.txt -rfc \
  -keystore config/keystore -storetype JKS
```

## 5.3.2 Configuring Oracle Directory Integration Platform for Oracle Unified Directory SSL

After configuring the Oracle Unified Directory (back-end directory) SSL communication, you must configure Oracle Directory Integration Platform.

Completing the following steps:

1. Import the trusted certificate that you have exported in [Configuring Oracle Unified Directory for SSL](#) into the Oracle Directory Integration Platform JKS (The Java Keystore you created in [Step 2](#)).

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

For example:

```
keytool -importcert -trustcacerts -alias OUD2 -file /home/Middleware/
  asinst_1/OUd/config/server-cert.txt -keystore /home/Middleware/dip.jks
```

The system will prompt for a keystore password. Type the password for this keystore (Enter the Java Keystore password that you created in [Step 5](#)).

2. Update the Oracle Directory Integration Platform SSL configuration, by running the following command:

UNIX

```
$ORACLE_HOME/bin/manageDIPServerConfig set -attribute sslmode -val 2 -h
  localhost -p 7005 -D "weblogic"
```

```
$ORACLE_HOME/bin/manageDIPServerConfig set -attribute backendhostport -val
  localhost:1636 -h localhost -p 7005 -D "weblogic"
```

Windows

```
ORACLE_HOME\bin\manageDIPServerConfig set -attribute sslmode -val 2 -h  
localhost -p 7005 -D "weblogic"
```

```
ORACLE_HOME\bin\manageDIPServerConfig set -attribute backendhostport -val  
localhost:1636 -h localhost -p 7005 -D "weblogic"
```

For more information, see [Arguments for manageDIPServerConfig](#).

You can also Log in to the Enterprise Manager 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 Unified Directory SSL port.

3. Restart the Oracle WebLogic managed server.

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

## 5.4 Verifying Oracle Directory Integration Platform

Verify the Oracle Directory Integration Platform installation using the `dipStatus` command, located in the `$ORACLE_HOME/bin/` directory.

### Note:

You must set the `WL_HOME` and `ORACLE_HOME` environment variables before executing the `dipStatus` and `dipConfigurator` commands.

The following is the syntax for the `dipStatus` command:

```
$ORACLE_HOME/bin/dipStatus -h localhost -p 7005 -D weblogic [-help]
```

- `-h` | `-host` identifies the Oracle WebLogic Server where Oracle Directory Integration Platform is deployed.
- `-p` | `-port` identifies the listening port of the Oracle Directory Integration Platform Managed Server.
- `-D` | `-wlsuser` identifies the 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.



You can also verify the Oracle Directory Integration Platform installation using the Oracle Enterprise Manager Fusion Middleware Control, as follows:

1. Open a browser, and access the Oracle Enterprise Manager Fusion Middleware Control using the following URL format:  
  
`http://hostname:port/em`
2. In the navigation panel on the left, click or expand **Identity and Access** and then select **DIP(12.2.1.3.0)**.
3. Click the DIP Server menu, point to **Administration**, and then select **Server Properties**.
4. Click **Test Connection** and verify the instance.

After you install and configure Oracle Directory Integration Platform , refer to the [Getting Started with Oracle Directory Integration Platform](#).

After configuring Oracle Unified Directory (back-end directory) non-SSL communication for Oracle Directory Integration Platform, you can synchronize or provision it with a connected directory, as described in [Synchronization Using Oracle Directory Integration Platform](#) or [Provisioning with the Oracle Directory Integration Platform](#).

# 6

## Configuring Oracle Internet Directory

You can configure Oracle Internet Directory as the back-end directory for Oracle Directory Integration Platform synchronization or provisioning.

### Topics:

- [Before You Configure Oracle Internet Directory as the Back-End Directory](#)
- [Configuring the Oracle WebLogic Server Domain for Oracle Directory Integration Platform with Oracle Internet Directory](#)
- [Configuring Oracle Directory Integration Platform for Oracle Internet Directory](#)
- [Verifying Oracle Directory Integration Platform](#)

### 6.1 Before You Configure Oracle Internet Directory as the Back-End Directory

Before configuring Oracle Internet Directory as the back-end directory, you must install Oracle Internet Directory and Oracle Directory Integration Platform.

- Install Oracle Internet Directory either as a collocated configuration or as a standalone configuration. For more information, see *Installing the Oracle Internet Directory Software* in *Installing and Configuring Oracle Internet Directory*.

#### Note:

You can only configure Oracle Internet Directory with Oracle Directory Integration Platform in SSL mode. The Oracle Internet Directory SSL port must be configured in SSL No Authentication Mode or SSL Server Authentication Only Mode.

- Configure Oracle Directory Integration Platform after you install Oracle Internet Directory binaries as described in *Configuring Oracle Directory Integration Platform* in *Installing and Configuring Oracle Internet Directory*.

### 6.2 Configuring the Oracle WebLogic Server Domain for Oracle Directory Integration Platform with Oracle Internet Directory

You must configure Oracle Directory Integration Platform with Oracle Internet Directory either in an existing or in a new WebLogic server Domain.

- [Configuring Oracle Directory Integration Platform with Oracle Internet Directory in an Existing WebLogic Domain](#)

- [Configuring Oracle Directory Integration Platform and Oracle Internet Directory in a New Oracle WebLogic Server Domain](#)

## 6.2.1 Configuring Oracle Directory Integration Platform with Oracle Internet Directory in an Existing WebLogic Domain

Perform the following steps to configure Oracle Directory Integration Platform with Oracle Internet Directory in an existing WebLogic administration domain:

### Note:

- During the Oracle Internet Directory domain configuration, if you have already selected the **Oracle Directory Integration Platform - 12.2.1.3.0[dip]** option in the Templates screen then you can skip this section.  
See *Selecting the Configuration Templates for Oracle Internet Directory in Oracle Fusion Middleware Installing and Configuring Oracle Internet Directory*.
- You must stop the Administration Server, Managed Servers, and Node Manager before updating the existing WebLogic domain.

1. Run the `ORACLE_HOME/oracle_common/common/bin/config.sh` script (UNIX) or `ORACLE_HOME\oracle_common\common\bin\config.cmd` (Windows).  
The **Configuration Type** screen is displayed.
2. Select **Update an existing domain**, and click **Next**.  
The **Templates** screen is displayed.
3. On the **Templates** screen, select **Update Domain Using Product Templates** and then select **Oracle Directory Integration Platform - 12.2.1.3.0[dip]** domain configuration option.

### Note:

When you select the **Oracle Directory Integration Platform - 12.2.1.3.0 [dip]** option, **Oracle Enterprise Manager 12.2.1.3.0 [em]** is automatically selected.

Click **Next**.

The **JDBC Data Sources** screen is displayed.

4. Make changes if required and then click **Next**  
The **JDBC Data Sources Test** screen is displayed.
5. Select the data sources to test, and click **Test Selected Connections**.  
Click **Next**.

The **Database Configuration Type** screen is displayed.

6. Make changes if required and then click **Get RCU Configuration** to retrieve the schema information. After successfully retrieving the schema information, click **Next** to continue.

The **JDBC Component Schema** screen is displayed.

7. Verify that the values populated are correct for all schemas and click **Next**.

The **JDBC Component Schema Test** screen is displayed.

8. You can select the component schema to test, and click **Test Selected Connections**. Wait for one or more connection tests to complete. If you do not want to test connections, deselect all data sources.

 **Note:**

In order to test connections, the database to which you are trying to connect must be running.

Click **Next**.

The **Advanced Configuration** screen is displayed.

9. Select **Managed Servers, Clusters, and Machines** option. Click **Next**.

The **Managed Servers** screen is displayed.

10. Specify the Managed Server name and click **Next**.

The **Clusters** screen is displayed.

11. Configure Clusters as required and click **Next**.

The **Machines** screen is displayed.

12. Select the **Machine** tab (for Windows) or **Unix Machine** tab. Click on **Add** and specify the machine name. Click **Next**.

13. If you added a machine on the Configure Machines screen, then the **Assign Servers to Machines** screen appears. On the Assign Servers to Machines screen, assign the Administration Server and the Managed server to the specified machine. Click **Next**.

14. On the **Configuration Summary screen**, review the domain configuration, and click **Update** to start extending the domain.

15. Click **Finish**, once the domain is extended.

Your existing Oracle Internet Directory domain is extended to support Oracle Directory Integration Platform.

## 6.2.2 Configuring Oracle Directory Integration Platform and Oracle Internet Directory in a New Oracle WebLogic Server Domain

Perform the configuration steps in this section only if you want to configure Oracle Directory Integration Platform and Oracle Internet Directory in a new Oracle WebLogic Server domain.

To configure Oracle Directory Integration Platform and Oracle Internet Directory in a new WebLogic domain:

1. Run the `ORACLE_HOME/oracle_common/common/bin/config.sh` script (UNIX) or `ORACLE_HOME\oracle_common\common\bin\config.cmd` (Windows).

The **Configuration Type** screen is displayed.

2. On the **Configuration Type** screen, select **Create a new domain** and enter the full path for the domain or use the **Browse** button to navigate to the directory in which your domains are located. Click **Next**.

The **Templates** screen is displayed.

3. On the **Templates** screen, make sure **Create Domain Using Product Templates** is selected, and then select the following templates:
  - **Oracle Directory Integration Platform - 12.2.1.3.0 [dip]**
  - **Oracle Internet Directory (Collocated) - 12.2.1.3.0 [oid]** (optional)

### Note:

When you select **Oracle Directory Integration Platform - 12.2.1.3.0 [dip]** option, the following components are automatically selected:

- **Oracle Enterprise Manager 12.2.1.3.0 [em]**
- **Oracle JRF - 12.2.1.3.0 [oracle\_common]**
- **Weblogic Coherence Cluster Extension 12.2.1.3 [wlserver]**

When you select **Oracle Internet Directory (Collocated) - 12.2.1.3.0 [oid]** option, then **Oracle Directory Services Manager - 12.2.1.3.0 [oid]** component is automatically selected.

Click **Next**.

Click The **Application Location** screen is displayed.

4. Click **Browse** and specify the full path to the directory in which you want to store the applications that are associated with the domain.

Click **Next**.

The **Administrator Account** screen is displayed.

5. Specify the user name and password for the default WebLogic Administrator account for the domain.

The password must be at least eight characters and must contain at least one number or special character. Confirm the password and click **Next**.

Make a note of these details as you will need them to start or restart the WebLogic domain in the following procedure.

The **Domain Mode and JDK** screen is displayed.

6. Specify the domain mode and Java Development Kit (JDK).
  - a. Select **Production** in the Domain Mode field.

 **Note:**

If you select **Production** mode as the domain, the node manager has a random username and password assigned to it. Use the WebLogic Server Administration Console to reset the password.

- b. Accept **Oracle Hotspot** as a default JDK location.
  - c. Click **Next**.

The **Database Configuration Type** screen is displayed.

7. Select RCU Data. This option instructs the Configuration Wizard to connect to the database's Service Table (STB) schema to automatically retrieve schema information for schemas needed to configure the domain.

 **Note:**

Ensure that you have created the database schemas required for Oracle Internet Directory. See *Creating the Database Schemas in Oracle Fusion Middleware Installing and Configuring Oracle Internet Directory*.

After selecting RCU Data:

- a. Enter the name of the server hosting the database in the **Host Name** field.

 **Note:**

Ensure that you do not specify localhost in the **Host Name** field.

- b. Enter the database DBMS name, or service name if you selected a service type driver in the **DBMS/Service** field.
  - c. Enter the port number on which the database listens.
  - d. Enter the username and password for connecting to the database's Service Table schema.
  - e. Click **Get RCU Configuration** to retrieve the schema information. After successfully retrieving the schema information, click **Next** to continue.

The **JDBC Component Schema** screen is displayed.

8. Verify that the values populated are correct for all schemas, and Click **Next**.

The **JDBC Component Schema Test** screen is displayed.

9. Test datasource connections that you just configured.

A green check mark in the Status column indicates a successful test. If you encounter issues, see the error message in the Connection Result Log section of the screen, fix the problem, then test the connection again.

The **Advanced Configuration** screen is displayed.

10. To complete domain configuration, select any of these options:

- **Administration Server:** Required to properly configure the Administration Server's listen address.
- **Node Manager:** Required to configure Node Manager.
- **Topology:** Required to configure the Managed Servers and cluster, and for configuring the machine and targeting Managed Servers to the machine.
- **Deployments and Services:** Required to target to servers or clusters.

Click **Next**.

11. Review each item on the **Configuration Summary** screen and verify that the information is correct.

To make any changes, go back to a screen by clicking the Back button or selecting the screen in the navigation pane. Domain creation does not start until you click **Create**.

A new WebLogic domain (for example: *base\_domain*) is created to support Oracle Directory Integration Platform and Fusion Middleware Control in the <ORACLE\_HOME>\user\_projects\domains directory (on Windows). On UNIX, the domain is created in the <ORACLE\_HOME>/user\_projects/domains directory.

## 6.3 Configuring Oracle Internet Directory (SSL) for Oracle Directory Integration Platform

Use the steps in the following order to configure Oracle Internet Directory (back-end directory) SSL communication for Oracle Directory Integration Platform.

- [Configuring Oracle Internet Directory for SSL](#)
- [Configuring Oracle Directory Integration Platform for Oracle Internet Directory SSL](#)
- [Adding Cipher Suites Configured for Oracle Internet Directory into Oracle Directory Integration Platform](#)

### 6.3.1 Configuring Oracle Internet Directory for SSL

Configure Oracle Internet Directory (back-end directory) in SSL mode. You can use the SSL No Authentication Mode or SSL Server Authentication Only Mode options to configure the SSL port.

 **Note:**

- Oracle recommends that you use SSL Server Authentication Only Mode option configured on an LDAPS port for Oracle Internet Directory.
- If Java Development Kit (JDK) 1.8.0\_201 or higher is installed on your system then the anonymous ciphers are disabled by default. If Oracle Internet Directory SSL is configured in SSL No Authentication Mode then you must enable the anonymous ciphers in the JDK by editing the `java.security` file (`JAVA_HOME/lib/security`) and removing `anon`, `NULL`, `DES`, and `3DES_EDE_CBC` from the `jdk.tls.disabledAlgorithms` security property.

See *Configuring Secure Sockets Layer (SSL) in Oracle Fusion Middleware Administering Oracle Internet Directory*.

## 6.3.2 Configuring Oracle Directory Integration Platform for Oracle Internet Directory SSL Authentication

After configuring the Oracle Internet Directory (back-end directory) SSL communication, you must configure Oracle Directory Integration Platform.

**Topics:**

- [Configuring Oracle Directory Integration Platform for Oracle Internet Directory SSL Server Authentication Only Mode](#)
- [Configuring Oracle Directory Integration Platform for Oracle Internet Directory SSL No Authentication Mode](#)

 **Note:**

Oracle recommends that you use SSL Server Authentication Only Mode option configured on an LDAPS port for Oracle Internet Directory.

### 6.3.2.1 Configuring Oracle Directory Integration Platform for Oracle Internet Directory SSL Server Authentication Only Mode

Complete the following steps to configure Oracle Directory Integration Platform to use Oracle Internet Directory SSL Server Authentication Only Mode:



 **Note:**

If you change the configuration from Oracle Internet Directory No Authentication Mode to SSL Server Authentication Only Mode, then you must delete the `TLS_DH_anon_WITH_AES_128_GCM_SHA256` and `SSL_DH_anon_WITH_3DES_EDE_CBC_SHA` cipher suites from Oracle Directory Integration Platform using the Oracle Fusion Middleware System MBean Browser.

This ensures that all the ciphers supported in Oracle Directory Integration Platform for the Java Development Kit (JDK) 1.8.0\_201 or higher are enabled.

1. Ensure that the Oracle WebLogic Administration Server and Oracle Directory Integration Platform managed server is running. If they are not running, then start as follows:

Administration Server:

```
DOMAIN_NAME/bin/startWebLogic.sh
```

 **Note:**

Where `DOMAIN_NAME` is the root directory of the domain. (The name of this directory is the name of the domain.). By default, this directory is `ORACLE_HOME\user_projects\domains\DOMAIN_NAME`.

Managed Server:

```
DOMAIN_NAME/bin/startManagedWebLogic.sh managed_server_name admin_url
```

See [Starting the Stack](#).

2. You must export the trusted certificate from the Oracle Internet Directory wallet using the `orapki` utility:
  - a. Ensure that the environment variables `JAVA_HOME` points to the Java installation directory.

```
JAVA_HOME=/usr/lang/JAVA/jdk1.8.0_131
export JAVA_HOME
```

- b. Ensure that the environment variables `ORACLE_HOME` for Oracle Internet Directory is set properly.

```
SETENV ORACLE_HOME <path to Oracle home location>
```

- c. 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:

```
$ORACLE_HOME/bin/orapki wallet export -wallet /home/Middleware/
Oracle_Home/oid/admin/oidwallet -dn "cn=oidhost.example.com,OU=My
Dept,O=My Company,L=Redwood City,ST=California,C=US " -cert oidcert.cer
```

3. Import the trusted certificate that you have exported to the `oidcert.cer` file into the Oracle Directory Integration Platform wallet:

```
keytool -importcert -trustcacerts -alias Some_alias_name -file
Path_to_certificate_file
```

For example:

```
keytool -importcert -trustcacerts -alias OID -file /home/Middleware/
oidcert.cer -keystore /home/Middleware/dip.jks
```

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

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

```
manageDIPServerConfig set -attribute keystorelocation -val
full_path_to_keystore -h weblogic_host -p weblogic_managed_server_port -D
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:

```
$ORACLE_HOME/bin/manageDIPServerConfig set -h localhost -p 7005 -D wlsuser
-attribute keystorelocation -val /home/Middleware/dip.jks
```

The system will prompt for the WebLogic password.

5. 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/oracle_common/common/bin/wlst.sh (UNIX) or
ORACLE_HOME\oracle_common\common\bin\wlst.cmd (Windows)
```

- 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")
```

6. Update the Oracle Directory Integration Platform SSL configuration, by running the following command:

### Unix

```
$ORACLE_HOME/bin/manageDIPServerConfig set -attribute sslmode -val 2 -h localhost -p 7005 -D "weblogic"
```

```
$ORACLE_HOME/bin/manageDIPServerConfig set -attribute backendhostport -val oidhost:3131 -h example.com -p 7005 -D "weblogic"
```

### Windows

```
ORACLE_HOME\bin\manageDIPServerConfig set -attribute sslmode -val 2 -h localhost -p 7005 -D "weblogic"
```

```
ORACLE_HOME\bin\manageDIPServerConfig set -attribute backendhostport -val oidhost:3131 -h example.com -p 7005 -D "weblogic"
```

For more information, see [Arguments for manageDIPServerConfig](#).

You can also Log in to the Enterprise Manager 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.

## 6.3.2.2 Configuring Oracle Directory Integration Platform for Oracle Internet Directory SSL No Authentication Mode

Complete the following steps to configure Oracle Directory Integration Platform to use Oracle Internet Directory No Authentication (SSL Mode 1) Mode:

### Note:

Oracle does not recommend using No Authentication (SSL Mode 1).

1. Ensure that the Oracle WebLogic Administration Server and Oracle Directory Integration Platform managed server is running. If they are not running, then start as follows:

Administration Server:

```
DOMAIN_NAME/bin/startWebLogic.sh
```

### Note:

Where `DOMAIN_NAME` is the root directory of the domain. (The name of this directory is the name of the domain.). By default, this directory is `ORACLE_HOME\user_projects\domains\DOMAIN_NAME`.

Managed Server:

```
DOMAIN_NAME/bin/startManagedWebLogic.sh managed_server_name admin_url
```

See [Starting the Stack](#).

2. Run the `manageDIPServerConfig` utility to update the Oracle Directory Integration Platform SSL configuration to use the Oracle Internet Directory SSL No Authentication Mode:

#### Unix

```
$ORACLE_HOME/bin/manageDIPServerConfig set -attribute sslmode -val 1 -h localhost -p 7005 -D "weblogic"
```

#### Windows

```
ORACLE_HOME\bin\manageDIPServerConfig set -attribute sslmode -val 1 -h localhost -p 7005 -D "weblogic"
```

For more information, see [Arguments for manageDIPServerConfig](#).

You can also Log in to the Enterprise Manager and update the Oracle Directory Integration Platform SSL configuration.

Choose **DIP > Server Properties**, then set SSL Mode to 1 and the port value to the Oracle Internet Directory SSL port.

## 6.3.3 Adding Cipher Suites Configured for Oracle Internet Directory into Oracle Directory Integration Platform

If the cipher suites configured for Oracle Internet Directory are not available or recognized in Oracle Directory Integration Platform then you must add those suites into Oracle Directory Integration Platform using the Oracle Fusion Middleware System MBean Browser.

For example, if Oracle Internet Directory SSL is configured in No Authentication Mode then by default anonymous ciphers are not recognized by Oracle Directory Integration Platform. Add the `TLS_DH_anon_WITH_AES_128_GCM_SHA256` and `SSL_DH_anon_WITH_3DES_EDE_CBC_SHA` cipher suites into Oracle Directory Integration Platform using the Oracle Fusion Middleware System MBean Browser.

To add cipher suites into Oracle Directory Integration Platform, complete the following steps:

### Note:

In a cluster environment, you must repeat the below steps for all of the Oracle Directory Integration Platform managed servers in the cluster.

1. Open a browser, and access the Fusion Middleware Control Console using the following URL format:
 

```
http://host1.example.com:7001/em
```
2. Enter the Oracle Fusion Middleware administrator user name and password and click **Login**.
3. From the target navigation pane, expand the domain.
4. From the domain home page, select the Managed Server (`wls_ods1`).

 **Note:**

The default value for Oracle Directory Integration Platform Managed Server is `wls_ods1`.

5. From the WebLogic Server menu, choose System **MBean Browser**. The **System MBean Browser** page is displayed.

 **Note:**

You can also click the **Find** icon to perform a search for an MBean, attribute. For example `com.bea:Name=wls_ods1,Type=Server`.

6. Expand **Configuration MBeans** in the MBean navigation tree and then select **com.bea > Server**.
7. Expand the Server node and then expand the Managed Server node (`wls_ods1`).
8. From the Managed Server node, expand SSL and then select the Managed Server MBEAN.

The **Configuration MBEAN** page is displayed.

9. Select **Attributes** tab and then select **CipherSuites**.

The **Attribute: Ciphersuites** page is displayed.

10. Click **Add** and then add the following ciphers:

- `TLS_DH_anon_WITH_AES_128_GCM_SHA256`
- `SSL_DH_anon_WITH_3DES_EDE_CBC_SHA`

 **Note:**

If Oracle Internet Directory is configured in No Authentication Mode, then you do not need to add any additional ciphers to Oracle Directory Integration Platform. However, if you have added other ciphers to Oracle Directory Integration Platform and also require the default ciphers for the JDK to be enabled, you can add those using the Oracle Fusion Middleware System MBean Browser.

For more information about cipher suites supported by Oracle Directory Integration Platform, see [Supported Out-of-Box Cipher Suites](#).

11. Click **Apply**.
12. Click **Activate Changes** in the Change Center.

 **Note:**

If you change the configuration from Oracle Internet Directory No Authentication Mode to SSL Server Authentication Only Mode, then you must delete the `TLS_DH_anon_WITH_AES_128_GCM_SHA256` and `SSL_DH_anon_WITH_3DES_EDE_CBC_SHA` cipher suites from Oracle Directory Integration Platform using the Oracle Fusion Middleware System MBean Browser.

This ensures that all the ciphers supported in Oracle Directory Integration Platform for the Java Development Kit (JDK) 1.8.0\_201 or higher are enabled.

## 6.4 Configuring Oracle Directory Integration Platform for Oracle Internet Directory

Use the `dipConfigurator` command to configure Oracle Directory Integration Platform for Oracle Internet Directory.

 **Note:**

Before running `dipConfigurator` to configure Oracle Internet Directory as the back-end directory, ensure that you have completed the following configuration based on the SSL implementation modes:

- No Authentication (SSL Mode 1):
  - [Configuring Oracle Directory Integration Platform for Oracle Internet Directory SSL No Authentication Mode](#)
  - [Adding Cipher Suites Configured for Oracle Internet Directory into Oracle Directory Integration Platform](#)
- SSL Server Authentication (SSL Mode 2): [Configuring Oracle Directory Integration Platform for Oracle Internet Directory SSL Server Authentication Only Mode](#)

Complete the following steps:

1. Set the `WL_HOME` and `ORACLE_HOME` environment variables for Oracle Directory Integration Platform.
2. Create the `dbconfigfile` file and name it as `db.properties` file.

The following shows an example of the `db.properties` file.

```
DRIVER_NAME:oracle.jdbc.OracleDriver
DRIVER_TYPE:thin
DB_HOST:myhost.us.example.com
DB_PORT:1521
```

```
DB_SID:orclpdb
DB_SERVICENAME: ORCLPDB.EXAMPLE.COM
```

3. Run the `dipConfigurator setup (<ORACLE_HOME>/bin)` command on the command line and enter the following arguments:




**Note:**

**Table 6-1 dipConfigurator Properties for Oracle Internet Directory**

Properties	Description
<code>wlshost</code>	Oracle WebLogic Server host name where Oracle Directory Integration Platform is deployed. The default host name is <code>localhost</code> .
<code>wlsport</code>	Listening port number of the Oracle WebLogic Administration Server where Oracle Directory Integration Platform is deployed. The default port number is <code>7001</code> .
<code>wlsuser</code>	Oracle WebLogic Server login user name.
<code>ldaphost</code>	Oracle Internet Directory host name. The default host name is <code>localhost</code> .
<code>ldapport</code>	Oracle Internet Directory server port number. The default value is <code>636</code> .
<code>isldapssl</code>	Accept the default value <code>true</code> for the Oracle Internet Directory configured as the back-end directory.
<code>ldapuser</code>	The bind DN to connect to the directory. The default value is <code>true</code> .
<code>isclustered</code> <BOOLEAN>	Specify if the Oracle Directory Integration Platform instance is in a cluster environment. The default value is <code>false</code> .

**Table 6-1 (Cont.) dipConfigurator Properties for Oracle Internet Directory**

Properties	Description
dbconfigfile	<p>The following property should be specified in the dbconfigfile file, and absolute path should be specified before running dipConfigurator setup.</p> <div style="border: 1px solid #0070C0; padding: 10px; margin: 10px 0;"> <p> <b>Note:</b></p> <p>For the database connection details and schemas, ensure that you provide the same value specified for the back-end Oracle Internet Directory installation and configuration. See <i>Creating the Database Schemas in Oracle Fusion Middleware Installing and Configuring Oracle Internet Directory</i>.</p> </div> <ul style="list-style-type: none"> <li>• DRIVER_NAME: Enter oracle.jdbc.OracleDriver.</li> <li>• DRIVER_TYPE: Enter thin.</li> <li>• DB_HOST: Host name of the machine on which the database is running. For example, exampledomain.com.</li> <li>• DB_PORT: Enter the port number for your database. The default port number for Oracle databases is 1521.</li> <li>• DB_SID: Specify the database SID. For example orcl.</li> <li>• DB_SERVICENAME: Enter the database service name . Example: orcl.exampledomain.com</li> <li>• URL: Enter the value only if you are using a Oracle RAC database.</li> </ul> <p>You must enter the Oracle RAC database connect string information in the short format:</p> <pre>hostname1:port1:instanceName1^hostname2:port2:instanceName2@serviceName</pre> <p>Example:</p> <pre>example1.com:1521:orcl1^example2.com:1521:orcl2@orcl.exampledomain.com</pre>
clustercheckinterval 1 <INT>	<p>Specify the frequency (milliseconds) at which an instance checks for server status (For example, detecting failed instances) with the other instances of the cluster. The default value is 120000 milliseconds.</p>

**Example:****Unix**

```
$ORACLE_HOME/bin/dipConfigurator setup -wlshost localhost -wlsport 7001 -wlsuser weblogic -ldaphost oidhost -ldapport 3131 -ldapuser "cn=orcladmin" -isldapssl true -dbconfigfile $ORACLE_HOME/db.properties
```

**Windows**

```
ORACLE_HOME/bin/dipConfigurator setup -wlshost localhost -wlsport 7001 -wlsuser weblogic -ldaphost oidhost -ldapport 3131 -ldapuser "cn=orcladmin" -isldapssl true -dbconfigfile ORACLE_HOME/db.properties
```



## 6.5 Verifying Oracle Directory Integration Platform

Verify the Oracle Directory Integration Platform installation using the `dipStatus` and `dipConfigurator` commands, located in the `$ORACLE_HOME/bin/` directory.

### Note:

You must set the `WL_HOME` and `ORACLE_HOME` environment variables before executing the `dipStatus` and `dipConfigurator` commands.

The following is the syntax for the `dipStatus` command:

```
$ORACLE_HOME/bin/dipStatus -h <hostName> -p <port> -D <wlsuser> [-ssl -keyStorePath <path> -keyStoreType <type>] [-help]
```

- `-h` | `-host` identifies the Oracle WebLogic Server where Oracle Directory Integration Platform is deployed.
- `-p` | `-port` identifies the listening port of the Oracle Directory Integration Platform Managed Server.
- `-D` | `-wlsuser` identifies the Oracle WebLogic Server login ID.
- `-ssl` executes the command in SSL mode.
- `keystorePath` identifies the full path to the keystore.
- `keyStoreType` identifies the type of the keystore identified by `-keystorePath`. For example: `-keystorePath jks` or `-keystorePath PKCS12`. The default value is `jks`.
- `-help` provides usage help for the command.

### 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.

After you install and configure Oracle Directory Integration Platform, refer to the [Getting Started with Oracle Directory Integration Platform](#).

After configuring Oracle Internet Directory (back-end directory) SSL communication for Oracle Directory Integration Platform, you can synchronize or provision it with a connected directory, as described in [Synchronization Using Oracle Directory Integration Platform](#) or [Provisioning with the Oracle Directory Integration Platform](#).

# 7

## Configuring Oracle Directory Server Enterprise Edition

You can configure Oracle Directory Server Enterprise Edition as the back-end directory for Oracle Directory Integration Platform synchronization.

### Topics:

- [Before You Configure Oracle Directory Server Enterprise Edition as the Back-End Directory](#)
- [Configuring Oracle Directory Server Enterprise Edition \(Non-SSL\) for Oracle Directory Integration Platform](#)
- [Configuring Oracle Directory Server Enterprise Edition \(SSL\) for Oracle Directory Integration Platform](#)
- [Verifying Oracle Directory Integration Platform](#)

### 7.1 Before You Configure Oracle Directory Server Enterprise Edition as the Back-End Directory

Ensure that you install Oracle Directory Integration Platform before you configure Oracle Directory Server Enterprise Edition as the back-end directory.

See [Configuring Oracle Directory Integration Platform](#) in *Oracle Fusion Middleware Installing and Configuring Oracle Internet Directory*.

### 7.2 Configuring Oracle Directory Server Enterprise Edition (Non-SSL) for Oracle Directory Integration Platform

Use the steps in the following order to configure Oracle Directory Server Enterprise Edition (back-end directory) non-SSL communication for Oracle Directory Integration Platform.

- [Installing and Configuring Oracle Directory Server Enterprise Edition](#)
- [Installing Oracle Directory Server Enterprise Edition Plug-In](#)
- [Creating Oracle Directory Server Enterprise Edition Suffixes](#)
- [Enabling the Retro Change Log for Oracle Directory Server Enterprise Edition](#)
- [Configuring the Oracle WebLogic Server Domain for Oracle Directory Integration Platform with Oracle Directory Server Enterprise Edition](#)
- [Starting the Oracle WebLogic Server and the Oracle Directory Server Enterprise Edition Instance](#)
- [Configuring Oracle Directory Integration Platform for Oracle Directory Server Enterprise Edition](#)

- [Adding Access Control Instructions \(ACIs\) for Oracle Directory Server Enterprise Edition](#)

## 7.2.1 Installing and Configuring Oracle Directory Server Enterprise Edition

Ensure that Oracle Directory Server Enterprise Edition is installed and configured.

See [Oracle Fusion Middleware Installation Guide for Oracle Directory Server Enterprise Edition](#) and [Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition](#).



### Note:

You must create a Directory Server instance, as described in [Creating Server Instances From Command Line](#) in *Oracle® Fusion Middleware Installation Guide for Oracle Directory Server Enterprise Edition*.

## 7.2.2 Installing Oracle Directory Server Enterprise Edition Plug-In

Install the Oracle Directory Server Enterprise Edition plug-in to synchronize the password for Oracle Directory Server Enterprise Edition.

You can synchronize the password for Oracle Directory Server Enterprise Edition, as described in [Configuring Password Synchronization for Oracle Directory Server Enterprise Edition](#). To install the Oracle Directory Server Enterprise Edition plug-in, complete the following steps:

1. Select the `dip-plugin.so` or `dip-plugin.dll` depending on your platform, from `ORACLE_HOME\dip\utils\dip-plugin (Windows)` or `$ORACLE_HOME/dip/utils/dip-plugin (UNIX)` directory in the Oracle Identity Management distribution package.
2. Copy it in:
  - 32-bit system: `INSTALL-PATH/dsee7/lib (UNIX)` or `INSTALL-PATH\dsee7\lib` or (Windows).
  - 64-bit system: `INSTALL-PATH/dsee7/lib64 (UNIX)`.

For more information about Oracle Directory Server Enterprise Edition plug-in, see [Oracle Fusion Middleware Developer's Guide for Oracle Directory Server Enterprise Edition](#).

## 7.2.3 Creating Oracle Directory Server Enterprise Edition Suffixes

Use the `dsconf` command to create the suffixes for the server's Directory Information Tree (DIT).

You must create the following suffixes:

- `cn=OracleContext`: This suffix is used for storing the Oracle Directory Integration Platform configuration details.

- Create a suffix for storing the synchronized data.

To create the suffixes, run the `dsconf` command on the command line:

```
dsconf create-suffix -i -c -h host -p port -D "cn=Directory Manager"
cn=OracleContext
dsconf create-suffix -i -c -h host -p port -D "cn=Directory Manager"
<data_suffix_name>
```

After creating the suffixes, you must do the following:

1. Create a `products.ldif` file with the following content for `cn=Products`:

```
dn: cn=Products,cn=OracleContext
changetype: add
cn: Products
objectclass: top
objectclass: extensibleObject
```

2. Run `ldapadd` command, to add the `cn=Products` entries:

```
ldapadd -h host -p port -D "cn=Directory Manager" -w <pwd> -v -f
products.ldif
```

For more information, see "[Creating Suffixes](#)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.

## 7.2.4 Enabling the Retro Change Log for Oracle Directory Server Enterprise Edition

Use the `dsconf` command to enable the retro change log for Oracle Directory Server Enterprise Edition.

Complete the following steps:

1. Run the `dsconf` command on the command line:

```
dsconf set-server-prop -h host -p port retro-cl-enabled:on
```

2. Restart the directory server instance, as described in "[Starting, Stopping, and Restarting a Directory Server Instance](#)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.

### Note:

Enabling the retro change log for Oracle Directory Server Enterprise Edition may impact Directory Server performance.

## 7.2.5 Configuring the Oracle WebLogic Server Domain for Oracle Directory Integration Platform with Oracle Directory Server Enterprise Edition

You must configure Oracle Directory Integration Platform with Oracle Directory Server Enterprise Edition either in an existing or in a new WebLogic server Domain.

- [Configuring Oracle Directory Integration Platform with Oracle Directory Server Enterprise Edition in an Existing WebLogic Domain](#)
- [Configuring Oracle Directory Integration Platform and Oracle Directory Server Enterprise Edition in a New Oracle WebLogic Server Domain](#)

### 7.2.5.1 Configuring Oracle Directory Integration Platform with Oracle Directory Server Enterprise Edition in an Existing WebLogic Domain

Perform the following steps to configure Oracle Directory Integration Platform with Oracle Directory Server Enterprise Edition in an existing WebLogic administration domain:

1. Run the `ORACLE_HOME/oracle_common/common/bin/config.sh` script (UNIX) or `ORACLE_HOME\oracle_common\common\bin\config.cmd` (Windows).  
The **Configuration Type** screen is displayed.
2. Select **Update an existing domain**, and click **Next**.  
The **Templates** screen is displayed.
3. On the **Templates** screen, select **Update Domain Using Product Templates** and then select **Oracle Directory Integration Platform - 12.2.1.3.0[dip]** domain configuration option.

 **Note:**

When you select the **Oracle Directory Integration Platform - 12.2.1.3.0 [dip]** option, **Oracle Enterprise Manager 12.2.1.3.0 [em]** is automatically selected.

Click **Next**.

The **JDBC Data Sources** screen is displayed.

4. Make changes if required and then click **Next**.  
The **JDBC Data Sources Test** screen is displayed.
5. Select the data sources to test, and click **Test Selected Connections**.  
Click **Next**.

The **Database Configuration Type** screen is displayed.

6. Make changes if required and then click **Get RCU Configuration** to retrieve the schema information. After successfully retrieving the schema information, click **Next** to continue.  
The **JDBC Component Schema** screen is displayed.
7. Verify that the values populated are correct for all schemas and click **Next**.  
The **JDBC Component Schema Test** screen is displayed.
8. You can select the component schema to test, and click **Test Selected Connections**. Wait for one or more connection tests to complete. If you do not want to test connections, deselect all data sources.

 **Note:**

In order to test connections, the database to which you are trying to connect must be running.

Click **Next**.

The **Advanced Configuration** screen is displayed.

9. Select **Managed Servers, Clusters, and Machines** option. Click **Next**.

The **Managed Servers** screen is displayed.

10. Specify the Managed Server name and click **Next**.

The **Clusters** screen is displayed.

11. Configure Clusters as required and click **Next**.

The **Machines** screen is displayed.

12. Select the **Machine** tab (for Windows) or **Unix Machine** tab. Click on **Add** and specify the machine name. Click **Next**.

13. If you added a machine on the Configure Machines screen, then the **Assign Servers to Machines** screen appears. On the Assign Servers to Machines screen, assign the Administration Server and the Managed server to the specified machine. Click **Next**.

14. On the **Configuration Summary** screen, review the domain configuration, and click **Update** to start extending the domain.

15. Click **Finish**, once the domain is extended.

Your existing domain is extended to support Oracle Directory Integration Platform.

## 7.2.5.2 Configuring Oracle Directory Integration Platform and Oracle Directory Server Enterprise Edition in a New Oracle WebLogic Server Domain

Run the Oracle Fusion Middleware Configuration Wizard to create a new Oracle WebLogic domain:

1. Run the `ORACLE_HOME/oracle_common/common/bin/config.sh` script (UNIX) or `ORACLE_HOME\oracle_common\common\bin\config.cmd` (Windows).

The **Configuration Type** screen is displayed.

2. On the **Configuration Type** screen, select **Create a new domain** and enter the full path for the domain or use the **Browse** button to navigate to the directory in which your domains are located. Click **Next**.

The **Templates** screen is displayed.

3. On the **Templates** screen, make sure **Create Domain Using Product Templates** is selected, and then select **Oracle Directory Integration Platform - 12.2.1.3.0 [dip]**.

 **Note:**

When you select **Oracle Directory Integration Platform - 12.2.1.3.0 [dip]** option, the following components are automatically selected:

- **Oracle Enterprise Manager 12.2.1.3.0 [em]**
- **Oracle JRF - 12.2.1.3.0 [oracle\_common]**
- **Weblogic Coherence Cluster Extension 12.2.1.3 [wlserver]**

Click **Next**.

Click The **Application Location** screen is displayed.

4. Click **Browse** and specify the full path to the directory in which you want to store the applications that are associated with the domain.

Click **Next**.

The **Administrator Account** screen is displayed.

5. Specify the user name and password for the default WebLogic Administrator account for the domain.

The password must be at least eight characters and must contain at least one number or special character. Confirm the password and click **Next**.

Make a note of these details as you will need them to start or restart the WebLogic domain in the following procedure.

The **Domain Mode and JDK** screen is displayed.

6. Specify the domain mode and Java Development Kit (JDK).
  - a. Select **Production** in the Domain Mode field.

 **Note:**

If you select **Production** mode as the domain, the node manager has a random username and password assigned to it. Use the WebLogic Server Administration Console to reset the password.

- b. Accept **Oracle Hotspot** as a default JDK location.
- c. Click **Next**.

The **Database Configuration Type** screen is displayed.

7. Select RCU Data. This option instructs the Configuration Wizard to connect to the database's Service Table (STB) schema to automatically retrieve schema information for schemas needed to configure the domain.

After selecting RCU Data:

- a. Enter the name of the server hosting the database in the **Host Name** field.

**Note:**

Ensure that you do not specify localhost in the **Host Name** field.

- b. Enter the database DBMS name, or service name if you selected a service type driver in the **DBMS/Service** field.
- c. Enter the port number on which the database listens.
- d. Enter the username and password for connecting to the database's Service Table schema.
- e. Click **Get RCU Configuration** to retrieve the schema information. After successfully retrieving the schema information, click **Next** to continue.

The **JDBC Component Schema** screen is displayed.

8. Verify that the values populated are correct for all schemas, and Click **Next**.

The **JDBC Component Schema Test** screen is displayed.

9. Test datasource connections that you just configured.

A green check mark in the Status column indicates a successful test. If you encounter issues, see the error message in the Connection Result Log section of the screen, fix the problem, then test the connection again.

The **Advanced Configuration** screen is displayed.

10. To complete domain configuration, select any of these options:

- **Administration Server:** Required to properly configure the Administration Server's listen address.
- **Node Manager:** Required to configure Node Manager.
- **Topology:** Required to configure the Managed Servers and cluster, and for configuring the machine and targeting Managed Servers to the machine.
- **Deployments and Services:** Required to target to servers or clusters.

Click **Next**.

11. Review each item on the Configuration Summary screen and verify that the information is correct.

To make any changes, go back to a screen by clicking the Back button or selecting the screen in the navigation pane. Domain creation does not start until you click **Create**.

A new WebLogic domain (for example: *base\_domain*) is created to support Oracle Directory Integration Platform and Fusion Middleware Control in the <ORACLE\_HOME>\user\_projects\domains directory (on Windows). On UNIX, the domain is created in the <ORACLE\_HOME>/user\_projects/domains directory.

## 7.2.6 Starting the Oracle WebLogic Server and the Oracle Directory Server Enterprise Edition Instance

After the WebLogic domain configuration is complete, you can start the servers to manage the domain.

After configuring the Oracle WebLogic Server domain, perform the following tasks:



1. Start the Administration Server, Node Manager and Managed Server as described in [Starting and Stopping the Oracle Stack](#).
2. Start the Oracle Directory Server Enterprise Edition instance:

```
$ dsadm start instance-path
```

## 7.2.7 Configuring Oracle Directory Integration Platform for Oracle Directory Server Enterprise Edition

After configuring the Oracle WebLogic Server domain, you must configure Oracle Directory Integration Platform for Oracle Directory Server Enterprise Edition.

Complete the following steps:

1. Set the `WL_HOME` and `ORACLE_HOME` environment variables for Oracle Directory Integration Platform.
2. Run the `dipConfigurator setup (<ORACLE_HOME>/bin)` command on the command line and enter the following arguments:

### Note:

If you are running the `dipConfigurator setup` for a secured LDAP (`isldapssl=true`) then you must configure Oracle Directory Integration Platform for Oracle Directory Server Enterprise Edition SSL communication mode. See [Configuring Oracle Directory Integration Platform for Oracle Directory Server Enterprise Edition SSL](#).

**Table 7-1 dipConfigurator Properties for Oracle Directory Server Enterprise Edition**

Properties	Description
<code>wlshost</code>	Oracle WebLogic Server host name where Oracle Directory Integration Platform is deployed.
<code>wlsport</code>	Listening port number of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed.
<code>wlsuser</code>	Oracle WebLogic Server login user name.
<code>ldaphost</code>	Oracle Directory Server Enterprise Edition host name.
<code>ldapport</code>	Oracle Directory Server Enterprise Edition server port number. The default value is 636.
<code>isldapssl</code>	Specify <code>true</code> or <code>false</code> , to enable or disable SSL.
<code>ldapuser</code>	The bind DN to connect to the Oracle Directory Server Enterprise Edition.
<code>isclustered</code> <BOOLEAN>	Specify if the Oracle Directory Integration Platform instance is in a cluster environment.
<code>clustercheckinterval</code> <INT>	Specify the frequency (milliseconds) at which an instance checks for server status (For example, detecting failed instances) with the other instances of the cluster.

**Example:****UNIX:**

```
$ORACLE_HOME/bin/dipConfigurator setup -wlshost localhost -wlsport 7001 -
wlsuser weblogic -ldaphost odseehost -ldapport 389 -ldapuser "cn=Directory
Manager" -isldapssl false
```

**Windows:**

```
ORACLE_HOME\bin\dipConfigurator setup -wlshost localhost -wlsport 7001 -
wlsuser weblogic -ldaphost odseehost -ldapport 389 -ldapuser "cn=Directory
Manager" -isldapssl false
```

**Note:**

You can synchronize the password for Oracle Directory Server Enterprise Edition, as described in [Configuring Password Synchronization for Oracle Directory Server Enterprise Edition](#).

## 7.2.8 Adding Access Control Instructions (ACIs) for Oracle Directory Server Enterprise Edition

Add the ACIs in an LDIF file for Oracle Directory Server Enterprise Edition using the `ldapmodify` command.

Run the `ldapmodify` command on the command line:

```
ldapmodify -h localhost -p 389 -D "cn=Directory Manager" -w secret12 <<EOF
dn: dc=<suffix_name>,dc=com
changetype: modify
add: aci
aci: (target="ldap:///dc=<suffix_name>,dc=com")(version 3.0; acl
"Entry-level DIP permissions"; allow (all,proxy)
groupdn="ldap:///cn=dipadmingrp,cn=DIPadmins,cn=Directory Integration
Platform,cn=Products,cn=oraclecontext"; allow (all,proxy)
groupdn="ldap:///cn=odipigroup,cn=DIPadmins,cn=Directory Integration
Platform,cn=Products,cn=oraclecontext"; )
-
add: aci
aci: (targetattr="*")(version 3.0; acl "Attribute-level DIP permissions";
allow (all,proxy) groupdn="ldap:///cn=dipadmingrp,cn=DIPadmins,cn=Directory
Integration Platform,cn=Products,cn=oraclecontext"; allow (all,proxy)
groupdn="ldap:///cn=odipigroup,cn=DIPadmins,cn=Directory Integration
Platform,cn=Products,cn=oraclecontext"; )
EOF
```

## 7.3 Configuring Oracle Directory Server Enterprise Edition (SSL) for Oracle Directory Integration Platform

Use the steps in the following order to configure Oracle Directory Server Enterprise Edition (back-end directory) SSL communication for Oracle Directory Integration Platform.

- [Configuring Oracle Directory Server Enterprise Edition for SSL](#)
- [Configuring Oracle Directory Integration Platform for Oracle Directory Server Enterprise Edition SSL](#)

### 7.3.1 Configuring Oracle Directory Server Enterprise Edition for SSL

Use the steps to configure Oracle Directory Server Enterprise Edition (back-end directory) SSL communication for Oracle Directory Integration Platform.

1. Configure Oracle Directory Server Enterprise Edition, as described in [Configuring Oracle Directory Server Enterprise Edition \(Non-SSL\) for Oracle Directory Integration Platform](#).
2. You can disable non-SSL communications so that the Oracle Directory Server Enterprise Edition communicates only through SSL. To do so, run the `dsconf` command on the command line:

UNIX:

```
$ dsconf set-server-prop -h host -P 1636 ldap-port:disabled
```

Windows:

```
dsconf set-server-prop -h host -P 1636 ldap-port:disabled
```

Restart the Oracle Directory Server Enterprise Edition instance:

```
dsadm restart instance-path
```

You can now no longer bind on the non secure port 389.

3. Create the Oracle Directory Integration Platform credentials, by doing the following:

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

```
$ORACLE_HOME/oracle_common/common/bin/wlst.sh (UNIX) or  
ORACLE_HOME\oracle_common\common\bin\wlst.cmd (Windows)
```

- b. Run the following command on the command line:

```
createCred(map="dip", key="jksKey", user="jksUser", password="password",  
desc="DIP SSL JKS")
```

Example:

```
createCred -wlshost localhost -wlsport 7001 -wlsuser  
weblogic -csfmap dip -csfkey jksKey -csfuser  
"cn=odisrv,cn=Registered Instances,cn=Directory Integration  
Platform,cn=products,cn=oraclecontext" -csfpassword password
```

For more information, see [Credential Store Framework for Oracle Directory Integration Platform](#).

4. Import the Oracle Directory Server Enterprise Edition certificate into Oracle Directory Integration Platform, by running the following command:

```
bin/dsadm show-cert -F der -o dsee-cert instance-path  
keytool -importcert -noprompt -trustcacerts -alias mycompany.com -file ~/  
dsee-cert -keystore ~/keystores/DIPKeyStore.jks -storepass <password>  
keytool -list -keystore ~/DIPKeyStore.jks -storepass <password>
```

## 7.3.2 Configuring Oracle Directory Integration Platform for Oracle Directory Server Enterprise Edition SSL

After configuring the Oracle Directory Server Enterprise Edition (back-end directory) SSL communication, you must configure Oracle Directory Integration Platform.

To configure Oracle Directory Integration Platform:

1. Update the Oracle Directory Server Enterprise Edition host name and port number, by running the `manageDIPServerConfig` utility (ORACLE\_HOME/bin directory):

```
manageDIPServerConfig set -h host -p port -D wlsuser -attribute {sslmode |
refreshinterval | quartzthreadcount | quartzdbretryinterval |
backendhostport |
keystorelocation} [-ssl -keystorePath PATH_TO_KEYSTORE -keystoreType TYPE]
[-value ATTRIBUTE_VALUE] [-help]
```

Example:

```
manageDIPServerConfig set -h localhost -p 7005 -D weblogic -attribute
backendhostport -val odseehost:1636
```

For more information, see [manageDIPServerConfig Utility](#).

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 ODSEE -file /home/Middleware/
asinst_1/ODSEE/admin/odseecert.txt -keystore /home/Middleware/dip.jks
```

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

### Note:

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

If you are using the Microsoft Active Directory as the connected directory, then you must do the following:

- a. Export the Microsoft Active Directory certificate, as described in [LDAP over SSL \(LDAPS\) Certificate](#).
- b. Import the Microsoft Active Directory certificate to the Oracle Directory Integration Platform, by running the following command:

```
keytool -importcert -noprompt -trustcacerts -alias mycompany.com -file
~/jpi-ad.cer -keystore ~/keystores/DIPKeyStore.jks -storepass <password>
```

- c. Verify the certificate, by running the following command:

```
keytool -list -keystore ~/keystores/DIPKeyStore.jks -storepass <password>
```

- d. Update the Keystore location in Oracle Directory Integration Platform:

```
$ORACLE_HOME/bin/manageDIPServerConfig set -attribute keystorelocation -val ~/keystores/DIPKeyStore.jks -h host -p 7005 -D "weblogic"
```

- e. Import the Microsoft Active Directory certificate to the Oracle Directory Server Enterprise Edition, by running the following command:

```
dsee7/bin/dsadm add-cert -C instance-path
```

- f. Restart the directory server instance:

```
dsadm restart instance-path
```

- g. Verify the certificate, by running the following command:

```
/dsee7/bin/certutil -L -d $INSTANCE/alias/ -P slapd-/dsee7/bin/certutil -L -d $INSTANCE/alias/ -P slapd- -n certAD/dsee7/bin/certutil -L -d $INSTANCE/alias/ -P slapd- -n host
```

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 localhost -p 7005 -wlsuser weblogic
```

The system will prompt for the WebLogic password.

4. Log in to the Enterprise Manager and update the Oracle Directory Integration Platform SSL configuration, by running the following command:

```
manageDIPServerConfig set -h weblogic_host -p weblogic_managed_server_port -wlsuser weblogic_user -attribute sslmode [-value ATTRIBUTE_VALUE] [-help]
```

For example:

```
manageDIPServerConfig set -h localhost -p 7005 -D weblogic -attribute sslmode -val 2
```

For more information, see [Arguments for manageDIPServerConfig](#).

5. Restart the Oracle WebLogic managed server.

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

## 7.4 Verifying Oracle Directory Integration Platform

Verify the Oracle Directory Integration Platform installation using the `dipStatus` command, located in the `$ORACLE_HOME/bin/` directory.

For more information, see [Verifying Oracle Directory Integration Platform](#).

After configuring Oracle Directory Server Enterprise Edition (back-end directory) non-SSL communication for Oracle Directory Integration Platform, you can synchronize it with a connected directory, as described in [Synchronization Using Oracle Directory Integration Platform](#).

# Part IV

## 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.

Part IV includes the following chapters:

- [Understanding the Oracle Directory Synchronization Service](#)
- [Configuring Directory Synchronization](#)
- [Managing Directory Synchronization Profiles](#)
- [Bootstrapping a Directory in Oracle Directory Integration Platform](#)
- [Synchronizing with Tables in Oracle Database](#)
- [Synchronizing with Oracle Human Resources](#)
- [Synchronizing with Third-Party Metadirectory Solutions](#)

# 8

## Understanding the Oracle Directory Synchronization Service

Know more about the synchronization profiles and connectors that link the Oracle back-end directory and connected directories.

### Topics:

- [Components Involved in Oracle Directory Synchronization](#)
- [Understanding How Synchronization Works](#)



### See Also:

[Introduction to Oracle Directory Integration Platform](#) for a conceptual discussion of Oracle Directory Integration Platform

## 8.1 Components Involved in Oracle Directory Synchronization

This topic discusses the components involved in Oracle Directory synchronization.

- [Overview of Connectors for Directory Synchronization](#)
- [Understanding Directory Synchronization Profiles](#)

### 8.1.1 Overview of 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.

- [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 connector provided with Oracle Directory Integration Platform 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.



- 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).

## 8.1.2 Understanding Directory Synchronization Profiles

A directory integration profile for synchronization, called a directory synchronization profile.

[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:**

LDAP Attribute Reference in *Oracle Fusion Middleware Reference for Oracle Identity Management* for a list and descriptions of the attributes in a directory integration profile.

## 8.2 Understanding How Synchronization Works

Learn more about how synchronization works between the Oracle back-end directory and a connected directory.

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

### Topics:

- [Understanding Synchronization from the Back-end Directory to a Connected Directory](#)
- [Understanding Synchronization from a Connected Directory to the Back-end Directory](#)

- [Understanding Directories Synchronization with Interfaces Not Supported by the Back-end Directory](#)

## 8.2.1 Understanding Synchronization 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:

- Retrieves the latest change log number up to which all changes have been applied.
- Checks each change log entry more recent than that number.
- Selects changes to be synchronized with the connected directory by using the filtering rules in the profile.
- 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 64 using the `manageSyncProfiles` command.

## 8.2.2 Understanding Synchronization 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.

## 8.2.3 Understanding Directories Synchronization 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.

# 9

## Configuring Directory Synchronization

This chapter explains how to configure directory synchronization and how to format mapping rules.

### Topics:

- [Registering Connectors in Oracle Directory Integration Platform](#)
- [About Synchronization Profile Templates](#)
- [Configure Connection Details for a Third-Party Directory](#)
- [Configuring Mapping Rules](#)
- [Extending Mappings Using Custom Plug-ins](#)
- [Configuring Matching Filters](#)
- [Location and Naming of Files](#)
- [Password Synchronization](#)

### See Also:

[Administering Oracle Directory Integration Platform](#) for information on using Oracle Enterprise Manager Fusion Middleware Control.

## 9.1 Registering Connectors in Oracle Directory Integration Platform

Before deploying a connector, you must 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](#)" for information about creating a directory synchronization profile using Oracle Enterprise Manager Fusion Middleware Control.

### See Also:

"[Understanding Directory Synchronization Profiles](#)"

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 2.16.840.1.113894.8 object identifier prefix is assigned to Oracle Directory Integration Platform platform-related classes and attributes.

The various synchronization profile entries in the directory are created under the following container:

```
cn=subscriber profile,cn=changelog subscriber,cn=Directory Integration Platform,cn=products,cn=oracleContext
```

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

```
orclodipagentname=OracleHRAgent,cn=subscriber profile,cn=changelog subscriber,cn=Directory Integration Platform,cn=products,cn=oracleContext
```

## 9.2 About Synchronization Profile Templates

When you install Oracle Directory Integration Platform, template profiles for synchronization are created for different types of directories.

For a complete list of supported directories, refer to the [Oracle Fusion Middleware Supported System Configurations](#) certification matrix.

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



### Note:

The synchronization profile template are examples. You must configure the mapping rules as described in [Configuring Mapping Rules](#), before using the template profile.

## 9.3 Configure Connection Details for a Third-Party Directory

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.

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.

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.

 **Note:**

If you are using Microsoft Active Directory as the third-party directory, 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 9-1](#):

**Table 9-1 Connection Detail Properties**

Property	Description
<code>odip.profile.condirurl</code>	The URL of the connected directory: <ul style="list-style-type: none"> <li>• To connect to an LDAP directory, use the form <i>host:port</i></li> <li>• To connect in SSL mode, use the form <i>host:port:1</i>.</li> <li>• To connect to a database, use the form <i>host:port.serviceName</i></li> </ul>
<code>odip.profile.condiraccount</code>	The DN or account name used to connect to the third-party directory

 **Note:**

- 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.

## 9.4 Configuring Mapping Rules

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

- [About Mapping Rules Attribute](#)

- [Distinguished Name Mapping](#)
- [About Domain Mapping Rules](#)
- [Domain Exclusion List](#)
- [Attribute-Level Mapping](#)
- [Attribute Exclusion List](#)
- [Manually Creating New Mapping Files](#)
- [Supported Attribute Mapping Rules and Examples](#)
- [Configuring Account Locking Synchronization](#)
- [Configuring Account Disabling Synchronization](#)
- [Example: Mapping File for a Tagged-File Interface](#)
- [Example: Mapping Files for an LDIF Interface](#)
- [Updating Mapping Rules](#)

### 9.4.1 About Mapping Rules Attribute

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 [Integrating with Novell eDirectory or OpenLDAP](#).



#### Note:

For information about configuring mapping rules that connect to Oracle Database, see [Updating the Configuration 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*.

```
DomainRules
srcDomainName1: [dstDomainName1]: [DomainMappingRule1]
srcDomainName2: [dstDomainName2]: [DomainMappingRule2]
[DomainExclusionList]
srcDomainForExclusion1
srcDomainforExclusion2

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.

## 9.4.2 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 9-2](#).

**Table 9-2 Domain Rule Components**

Component Name	Description
SrcDomainName	Name of the domain or container of interest. Specify NONLDAP for sources other than LDAP and LDIF.
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 <code>SrcDomainName</code> is an LDAP or LDIF domain, then this field assumes the same value. However, if the value assigned to <code>SrcDomainName</code> 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 <code>SrcDomainName</code> 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. For more information, see <a href="#">About Domain Mapping Rules</a> .

### Example 9-1 Example of Distinguished Name Mapping

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

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

### Example 9-2 Example of One-to-One Distinguished Name Mapping

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

- The source 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 destination 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 source directory matches the DN in the destination 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 subtree `cn=groups,dc=us,dc=mycompany,dc=com` from the source directory under the entry `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
```

### 9.4.3 About Domain Mapping Rules

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.

#### **dnpart Attribute**

You use the `dnpart` attribute as a variable to store a string and then you can reuse the variable in a domain mapping rule.

For the `rdn` attribute, a comma is part of the attribute and in the `dnpart` attribute, the comma is used as a separator to create extra levels in the directory tree.

To enable virtual distinguished name mapping, the domain mapping rule is as follows:

```
dc=source,dc=com:dc=destination,dc=com:cn=*,dnpart,dc=destination,dc=com
```

You must edit the attribute mapping rules as shown in the following example:

```
l:1: :europeanPerson:dnpart: : "cn=" + l + ",cn=europe"
l:1: :americanPerson:dnpart: : "cn=" + l + ",cn=america"
```

In the above example, an `ObjectClass` is used to define the creation of the DN with the extra level `cn=europe` or `cn=america`.

### multivalued Attribute

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
telephoneNumber:1 : : :telephonenumber: :person
address:1 : : :postaladdress: :person
address:1 : : :postaladdress: :person
address:1 : : :postaladdress: :person
state: : : :st: :locality
street:1 : : :street: :locality
zip: : : :postalcode: :locality
```

## 9.4.4 Domain Exclusion List

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 DN of the containers in the source directory.

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

```
DomainExclusionList
cn=sales,cn=users,dc=us,dc=mycompany,dc=com
cn=marketing,cn=users,dc=us,dc=mycompany,dc=com
```

**Example 9-3** shows an example map file that includes the `DomainExclusionList` header. In this example, the entries under `cn=sales,cn=users,dc=us,dc=mycompany,dc=com` and `cn=marketing,cn=users,dc=us,dc=mycompany,dc=com` will be excluded.

## 9.4.5 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 9-3](#).



### Note:

You can specify the literals using single quotation marks ( ' ') or with double quotation marks ( " "). Ensure that a single character is enclosed by single quotation marks ( ' ') and multi character literals are enclosed by double quotation marks ( " ").

**Table 9-3 Components in Attribute Rules**

Component Name	Description
<code>SrcAttrName</code>	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 <code>ColumnName</code> in the table specified by the <code>SrcObjectClass</code> . For other repositories this parameter can be appropriately interpreted.
<code>ReqAttrSeq</code>	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.
<code>SrcAttrType</code>	This parameter refers to the attribute type—for example, integer, string, binary—that validates the mapping rules.
<code>SrcObjectClass</code>	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.

**Table 9-3 (Cont.) Components in Attribute Rules**

Component Name	Description
DstAttrName	<p>Optional attribute. If it is not specified, then the SrcAttrName is assumed.</p> <p>For LDAP-compliant directories, this parameter refers to the name of the attribute at the destination.</p> <p>For Oracle Database repositories, it refers to the ColumnName in the table specified by the SrcObjectClass.</p> <p>For other repositories, this parameter can be appropriately interpreted.</p> <p>If you specify "-" and leave the DstAttrType and DstObjectClass fields empty then the value computed by the mapping function will not be stored in any attribute at the destination entry. This is useful for mapping functions that execute tasks but do not return values (For example, AccountLockOut())</p>
DstAttrType	<p>This parameter refers to the attribute type. It supports all the attributes types that is supported by the Oracle Internet Directory syntax. You must ensure compatibility of the source and destination attribute types. Oracle Directory Integration Platform does not ensure this compatibility.</p>
DstObjectClass	<p>If it is not specified, then the SrcObjectClass value is used.</p> <p>For LDAP-compliant directories, this parameter refers to the object class to which the attribute belongs, and is optional.</p> <p>For Oracle Database repositories, it refers to the table name, and is mandatory.</p> <p>For other repositories this parameter may be ignored.</p>
AttrMapping Rule	<p>Optional arithmetic expression with the following operators (+ and ) and functions.</p> <p>For more information, see "<a href="#">Supported Attribute Mapping Rules and Examples</a>".</p> <p>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 (").</p>

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.

## 9.4.6 Attribute Exclusion List

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 9-3** shows an example map file that includes both the `DomainExclusionList` and `AttributeExclusionList` headers. In this example, the entries under `cn=sales,cn=users,dc=us,dc=mycompany,dc=com` and `cn=marketing,cn=users,dc=us,dc=mycompany,dc=com` will be excluded, and all filtered entries will exclude (not contain) the `facsimileTelephoneNumber` and `telephonenumber` attributes.

### Example 9-3 Example Map File Using DomainExclusionList and AttributeExclusionList Headers

```
DomainRules
cn=users,dc=us,dc=mycompany,dc=com : ou=people,dc=us,dc=myothercompany,dc=com:
DomainExclusionList
cn=sales,cn=users,dc=us,dc=mycompany,dc=com
cn=marketing,cn=users,dc=us,dc=mycompany,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
r(trunc1(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 Kerberos principalname
userPrincipalName: : :user:krbPrincipalName: :orcluserv2:trunc(userPrincipalName,
'@')+ '@'+toupper(trunc1(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
```

## 9.4.7 Manually Creating New Mapping Files

This section describes how to create mapping files manually without using Oracle Enterprise Manager Fusion Middleware Control.

 **Note:**

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](#).

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 `cn=Anne Smith` to many attributes.



#### See Also:

The mapping file examples at the end of this chapter

## 9.4.8 Supported Attribute Mapping Rules and Examples

This section lists the supported attribute mapping rules.

- 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 and removes the first occurrence of the specified char and everything that appears on the right side of the char. For example:

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

For example, if `mail` is `John.Doe@example.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 and removes the first occurrence of the specified char and everything that appears on the left side of the char. For example:

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

- `truncr(str, char)`: Truncates the string and removes the first occurrence of the specified char and everything that appears on the right side of the 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=testuser1,cn=dstdomain`.

- `substring(String src,String startPosition,String endPosition)`: To return the substring starting at the <start position> and ending at the <end position> or until the end of the string, if it is not provided.

```
sn: : : person: employeenumber: : inetorgperson:substring(sn,"2","5")
sn: : : person: givenname: : inetorgperson:substring(sn,"3")
```

- **Literals:**

```
Userpassword: : :person: userpassword: :person: 'password'
```

- `fsptodn`: To compute a DN from a FSP (Foreign Security Principals). For example:

```
fsp : : : : dn : : top : fsptodn(fsp)
```

- **AccountLockOut:** To lock user accounts after multiple failed bind attempts. For example:  

```
pwdAccountLockedTime:::-::AccountLockOut(pwdAccountLockedTime,<lockout-duration>,<bind-failures>)
```

  
For more information, see [Configuring Account Locking Synchronization](#).
- **AccountDisable:** To temporarily disable a user's account, then enable it again. For example, if using Oracle Unified Directory as the source and destination directory:  

```
ds-pwp-account-disabled:1::ds-pwp-account-disabled::top:
```

  
For more information, see [Configuring Account Disabling Synchronization](#).
- **OnDemandPassword:** To synchronize the password from the connected directory to the back-end directory by configuring an import profile. For example:  

```
pwdLastSet : : : user : orclODIPPwdLastSet : : top :  
onDemandPassword(pwdLastSet)
```

  
For more information, see [Password Synchronization](#).
- **PasswordTranslate:** To synchronize the password from the back-end directory to the connected directory by configuring an export profile. For example,  

```
orclodiptranslatepassword: : : unicodepwd : : user :  
passwordtranslate(orclodiptranslatepassword)
```

  
For more information, see [Password Synchronization](#).

## 9.4.9 Configuring Account Locking Synchronization

You can use the back-end directory or connected directory account lockout feature to lock user accounts after too many failed bind attempts. Once an account has been locked, that user will not be allowed to authenticate. The lockout may be temporary (automatically ending after a specified period of time) or permanent (remaining in effect until an administrator resets the user's password).

If the back-end directory and the associated connected directories are synchronized with Oracle Directory Integration Platform, and if an account on one directory is locked out then the Oracle Directory Integration Platform will also lock out the account for other directory.



### Note:

Account lockout synchronization in Oracle Directory Integration Platform is supported only for Oracle Unified Directory and Oracle Directory Server Enterprise Edition back-end directories.

### Topics:

- [Prerequisites for Account Lockout Synchronization](#)
- [Enabling Account Lockout Synchronization](#)

### 9.4.9.1 Prerequisites for Account Lockout Synchronization

Complete the following before configuring account lockout synchronization.

- Ensure that the back-end directory and the connected directory are time synchronized. Refer to your operating system-specific documentation for more information on time synchronization.
- Ensure that the password policies for the back-end directory and the connected directory is configured to have the same properties like enabling or disabling account lockout, bind failures, and lockout duration.

For more information, see

- ["Managing the Default Password Policy"](#) in the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.
- ["Managing Password Policies"](#) in the *Oracle Fusion Middleware Administering Oracle Unified Directory*.

#### Note:

If the password policies are not assigned to an entry, then the default value is assigned. Ensure that the source directory and the connected directory are configured to have the same default password policies.

### 9.4.9.2 Enabling Account Lockout Synchronization

This section describes how to enable account lockout synchronization for Oracle Unified Directory, Oracle Directory Server Enterprise Edition, or Microsoft Active Directory source directory.

You can enable account lockout synchronization, by editing the mapping rule as follow:

#### **When the Source Directory is Oracle Unified Directory or Oracle Directory Server Enterprise Edition**

To enable account lockout synchronization:

```
pwdAccountLockedTime:::-::AccountLockOut(pwdAccountLockedTime,  
<lockout-duration>, <bind-failures>)
```

`bind-failures`: The number of failed binds after which an account can be locked out by Oracle Directory Integration Platform for the destination directory.

`lockout-duration`: The time that was configured for the password policies for automatic unlocking after an account is automatically locked. Ensure that the password policies in the source and destination directories have the same duration.

#### **Example:**

```
pwdAccountLockedTime:::-::AccountLockOut(pwdAccountLockedTime, "300", "3")
```

### When the Source Directory is Microsoft Active Directory

You can add the following mapping rule:

```
lockoutTime,msds-user-account-control-
computed:::user:-:::AccountLockOut(msds-user-account-control-
computed,<bind-failures>)
```

`bind-failures`: The number of failed binds after which an account can be locked out by Oracle Directory Integration Platform for the destination directory.

#### Example:

```
lockoutTime,msds-user-account-control-
computed:::user:-:::AccountLockOut(msds-user-account-control-
computed,"3")
```

## 9.4.10 Configuring Account Disabling Synchronization

If the Administrator disables an account, then you cannot login until the Administrator enables it. You can use the back-end directory or the connected directory feature to temporarily disable a user's account, then enable it again. If the account is temporarily disabled or enabled in one directory then Oracle Directory Integration Platform will temporarily disable or enable the account in the associated directory.

You can configure account disable synchronization, by adding the mapping rules described in [Table 9-4](#):



#### Note:

Account disabling synchronization in Oracle Directory Integration Platform is supported only for Oracle Unified Directory and Oracle Directory Server Enterprise Edition back-end directories.

**Table 9-4 Mapping Attribute for Account Disabling Synchronization**

Source Directory	Destination Directory	Mapping Rules
Oracle Unified Directory	Oracle Unified Directory	<code>ds-pwp-account-disabled:1:::ds-pwp-account-disabled::top:</code>
Oracle Directory Server Enterprise Edition	Oracle Unified Directory	<code>nsAccountLock:1:::ds-pwp-account-disabled::top:</code>
Microsoft Active Directory	Oracle Unified Directory	<code>userAccountControl:1:::ds-pwp-account-disabled::top:</code> <code>AccountDisable(userAccountControl)</code>
Oracle Unified Directory	Oracle Directory Server Enterprise Edition	<code>ds-pwp-account-disabled:1:::nsAccountLock::top:</code>

Table 9-4 (Cont.) Mapping Attribute for Account Disabling Synchronization

Source Directory	Destination Directory	Mapping Rules
Oracle Directory Server Enterprise Edition	Oracle Directory Server Enterprise Edition	nsAccountLock:1::: nsAccountLock::top:
Microsoft Active Directory	Oracle Directory Server Enterprise Edition	userAccountControl:1:::nsAccountLock::top: AccountDisable(userAccountControl)
Oracle Unified Directory	Microsoft Active Directory	modifiersName,ds-pwp-account-disabled:1:::userAccountControl: :user:AccountDisable(ds-pwp-account-disabled, "544") Oracle Directory Integration Platform uses 544 as the default value for the userAccountControl attribute during bootstrapping. The ADS_UF_ACCOUNTDISABLE attribute will be set to reflect the disabled state in the source directory. For more information on userAccountControl attribute, refer to the Microsoft Active Directory documentation.
Oracle Directory Server Enterprise Edition	Microsoft Active Directory	modifiersName,nsAccountLock:1::: userAccountControl::user:AccountDisable(nsAccountLock, "544") Oracle Directory Integration Platform uses 544 as the default value for the userAccountControl attribute during bootstrapping. The ADS_UF_ACCOUNTDISABLE attribute will be set to reflect the disabled state in the source directory. For more information on userAccountControl attribute, refer to the Microsoft Active Directory documentation.

 **Note:**

To disable the user in the source directory, see:

- [Oracle Fusion Middleware Administering Oracle Unified Directory](#)
- [Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition](#)

## 9.4.11 Example: Mapping File for a Tagged-File Interface

This example describes 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=metahr,dc=com:cn=%,dc=metahr,dc=com
AttributeRules
firstname: : : :cn: :person
lastname: : : :sn: :person
lastname: : : :cn: :person
email: : : :cn: :person: trunc(email,'@')
firstname,lastname: : : :cn: :person: firstname+", "+lastname
firstname,lastname: : : :cn: :person: lastname+", "+firstname
EmployeeNumber: : : :employeenumber: :inetOrgperson
EMail: : : :mail: :inetOrgperson
TelephoneNumber1: : : :telephonenumber: :person
TelephoneNumber2: : : :telephonenumber: :person
TelephoneNumber3: : : :telephonenumber: :person
Address1: : : :postaladdress: :person:Address1 | Address2
Address2: : : :Telephonenumber: :person:
state: : : :st: :locality
street1: : : :street: :locality
zip: : : :postalcode: :locality
town_or_city: : : :l: :locality
Title: : : :title: :organizationalperson
#Sex: : : :sex: :person
#socialsecurity: : : :ssn: :person
country: : : :c: :country
#BirthDate: : : :birthday: :organizationalperson
employeenumber:1: : :userpassword: :person: "*****"+employeenumber
```



### Note:

[About Domain Mapping Rules](#) for more information.

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,'@')
```

## 9.4.12 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 create the sample integration profiles are located in the `$ORACLE_HOME/ldap/odi/samples` directory.

### Note:

See [Configure 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.

## 9.4.13 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.

### Topics:

- [Adding an Entry to the Mapping Rules File](#)
- [Modifying an Entry in the Mapping Rules File](#)
- [Deleting an Entry from the Mapping Rules File](#)

### 9.4.13.1 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:
```

### 9.4.13.2 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.

### 9.4.13.3 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" 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/>

## 9.5 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.

### Topics:

- [Writing Custom Plug-Ins](#)
- [Understanding Mapping Plug-In Evaluation Constraints](#)
- [Adding Mapping Plug-Ins](#)
- [Applications of Mapping Plug-Ins](#)
- [Example Plug-In Usage](#)

### 9.5.1 Writing Custom Plug-Ins

This section describes how to extend mapping functionality using custom plug-ins.

To extend mapping functionality using custom plug-ins you must implement a class in the `oracle.ldap.odip.util.mapapi.plugin` package and implement the `oracle.ldap.odip.util.mapapi.IMapOperation` interface, as follows:

```
package oracle.ldap.odip.util.mapapi;

import java.util.Vector;

public interface IMapOperation
{
    public Vector evaluate(Vector operands);
}
```

The `operands` argument received by the `evaluate()` method 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.

 **Note:**

The `Vector evaluate` method can also return an empty vector (It cleans or deletes the attribute) or a null value (It keeps the current values unmodified).

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"

For example:

```
package oracle.ldap.odip.util.mapapi.plugin;

import java.util.Vector;

public class MyPlugin
    implements oracle.ldap.odip.util.mapapi.IMapOperation
{
    public Vector evaluate(Vector operands)
    {
        ...
    }
}
```

## 9.5.2 Understanding Mapping Plug-In Evaluation Constraints

This section explains the 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) + givenaname
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)
```

### 9.5.3 Adding Mapping Plug-Ins

You can add a mapping plug-in to Oracle Directory Integration Platform by copying the mapping plug-in JAR file.

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

1. If it is running, stop the Oracle WebLogic Managed Server hosting Oracle Directory Integration Platform. See [Stopping the Stack](#).
2. Copy the mapping plug-in JAR file to the `/APP-INF/lib/` directory in the Oracle Directory Integration Platform Managed Server. For example:

```
ORACLE_HOME/user_projects/domains/DOMAIN_NAME/servers/  
MANAGED_SERVER_NAME/tmp/_WL_user/DIP_VERSION_NUMBER/RANDOM_CHARACTERS/APP-  
INF/lib/
```

3. Start the Oracle WebLogic Managed Server hosting Oracle Directory Integration Platform. See [Starting the Stack](#).

### 9.5.4 Applications of Mapping Plug-Ins

This section describes various applications of Mapping plug-ins.

#### Topics:

- [Support for New Mapping Operations](#)
- [Support for Multiple Literal Values](#)

#### 9.5.4.1 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)
```

### 9.5.4.2 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.

## 9.5.5 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;
}

```

## 9.6 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](#)

## 9.6.1 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))))
```

## 9.6.2 Filtering Changes from a Change Log

This section describes the operators, which are provided by Oracle Directory Integration Platform, to specify a matching filter.

For connected directories that store changes in a change log, you can use the following simple operators 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=subscriberprofile,cn=changelog
subscriber,cn=products,cn=oraclecontext)
(modifiersname=orclodipagentname=imp2,cn=subscriberprofile,cn=changelog
subscriber,cn=products,cn=oraclecontext)))
```

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.

## 9.7 Location and Naming of Files

This section 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 9-5 Location and Names of Files**

File	File Name
Import data file	<code>\$ORACLE_HOME/ldap/odi/data/import/Profile_Name.dat</code>
Export data file	<code>\$ORACLE_HOME/ldap/odi/data/export/Profile_Name.dat</code>

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

## 9.8 Password Synchronization

Using the Oracle Directory Integration Platform password synchronization functionality, you can maintain a single password across the back-end directory and the connected directory.

### Topics:

- [Understanding Password Synchronization Mechanism](#)
- [Configuring Password Synchronization for Oracle Unified Directory](#)
- [Configuring Password Synchronization for Oracle Directory Server Enterprise Edition](#)
- [Configuring Password Synchronization for Oracle Internet Directory](#)

### Note:

Oracle recommends using the SSL connection to synchronize the password for the back-end directory and the connected directory.

### 9.8.1 Understanding Password Synchronization Mechanism

Password synchronization differs depending on the back-end directory and the connected directory due to different configuration mode and deployment mode. Oracle Directory Integration Platform provides several mechanisms to synchronize passwords depending on the selected back-end directory and connected directory.

Oracle Directory Integration Platform supports the following password mechanism:

- Enabling the Oracle Internet Directory Password policy: To synchronize passwords from Oracle Internet Directory to a non Oracle Internet Directory connected directory, you must enable the password policy and you may have to enable reversible password encryption in the Oracle Internet Directory server. For more information, see [Enable Password Synchronization from the Oracle Back-end Directory to a Connected Directory](#).
- Delegated Authentication: External authentication plug-ins, such as the Microsoft Active Directory external authentication plug-in, are available for the back-end directory and enable users to log in to the Oracle environment by using their Microsoft Windows credentials.
  - Oracle Internet Directory External Authentication Plug-ins: To synchronize passwords from Oracle Internet Directory to a non Oracle Internet Directory connected directory using Java-based external authentication plug-ins. For more information, see [Configuring External Authentication Plug-ins](#).
  - Pass-Through Authentication: Pass-through authentication (PTA) is a mechanism by which bind requests are filtered by bind DN. One Directory Server (the delegator) receives the bind request and, based on the filter, can consult another Directory Server (the delegate) to authenticate bind requests. As part of this functionality, the PTA plug-in enables the delegator Directory Server to accept simple password-based bind operations for entries that are not necessarily stored in its local database. A typical scenario for pass-through authentication involves passing authentication through to Active Directory for users coming from Oracle Unified Directory or Oracle Directory Server Enterprise Edition. For more information, see:
    - \* The section "Understanding Pass-Through Authentication" in the *Oracle Fusion Middleware Administering Oracle Unified Directory*.
    - \* The section "[Pass-Through Authentication](#)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.
- Oracle Password Filter: To synchronize password only from Microsoft Active Directory to a back-end directory using the Oracle Password Filter. For more information, see [Deploying the Oracle Password Filter for Microsoft Active Directory](#).
- Synchronization using password hashing techniques: To synchronize password from Oracle Directory Integration Platform to Oracle Unified Directory or Oracle Directory Server Enterprise Edition.

Oracle Unified Directory and Oracle Directory Server Enterprise Edition support the same set of password hashing techniques. To synchronize passwords between Oracle Directory Integration Platform and Oracle Directory Server Enterprise Edition:

- Ensure that SSL server authentication mode is configured for both directories, as described in [Configuring Oracle Unified Directory \(SSL\) for Oracle Directory Integration Platform](#).
- Ensure that the following mapping rule exists in the mapping file:

```
Userpassword: : :person:userpassword: :person
```

For more information on Oracle Directory Server Enterprise Edition hashing synchronization, see [Understanding How to Synchronize Passwords for Oracle Directory Server Enterprise Edition](#) .



- On-Demand Password: To synchronize the password from a connected directory to Oracle Unified Directory or Oracle Directory Server Enterprise Edition back-end directories. You can synchronize the password from the connected directory to the back-end directory by configuring an import profile. For more information, see:
  - [Synchronizing Password from a Connected Directory to the Oracle Unified Directory using On-Demand Password](#)
  - [Synchronizing Password from a Connected Directory to the Oracle Directory Server Enterprise Edition using On-Demand Password](#)
- Password Translate: To synchronize the password from a Oracle Unified Directory or Oracle Directory Server Enterprise Edition back-end directory to a connected directory. You can synchronize the password from the back-end directory to the connected directory by configuring an export profile. For more information, see:
  - [Synchronizing Password from the Oracle Unified Directory to a Connected Directory using Password Translate](#)
  - [Synchronizing Password from the Oracle Directory Server Enterprise Edition to a Connected Directory Using Password Translate](#)

Table 9-6 lists the password synchronization mechanism options for the Oracle Directory Integration Platform supported directory server.

**Table 9-6 Password Synchronization Mechanism**

Source	Destination	Supported Password Synchronization Mechanisms
Oracle Unified Directory	Oracle Unified Directory	<ul style="list-style-type: none"> <li>• Oracle Unified Directory (Destination directory) pass-through authentication</li> <li>• On-Demand Password</li> <li>• Password Translate</li> <li>• Synchronization using password hashing techniques</li> </ul>
Oracle Unified Directory	Oracle Internet Directory	<ul style="list-style-type: none"> <li>• Oracle Internet Directory External Authentication Plug-ins</li> <li>• Password Translate</li> <li>• Synchronization using password hashing techniques</li> </ul>
Oracle Unified Directory	Oracle Directory Server Enterprise Edition	<ul style="list-style-type: none"> <li>• Oracle Directory Server Enterprise Edition (Destination Directory) pass-through authentication</li> <li>• On-Demand Password</li> <li>• Password Translate</li> <li>• Synchronization using password hashing techniques</li> </ul>
Oracle Unified Directory	Microsoft Active Directory	<ul style="list-style-type: none"> <li>• Password Translate</li> </ul>
Oracle Internet Directory	Oracle Unified Directory	<ul style="list-style-type: none"> <li>• Enabling the Oracle Internet Directory password policy.</li> <li>• Oracle Unified Directory (Destination Directory) pass-through authentication</li> <li>• Synchronization using password hashing techniques</li> </ul>

Table 9-6 (Cont.) Password Synchronization Mechanism

Source	Destination	Supported Password Synchronization Mechanisms
Oracle Internet Directory	Oracle Internet Directory	<ul style="list-style-type: none"> <li>Enabling the password policy.</li> <li>Oracle Internet Directory External Authentication Plug-ins</li> <li>Synchronization using password hashing techniques</li> </ul>
Oracle Internet Directory	Oracle Directory Server Enterprise Edition	<ul style="list-style-type: none"> <li>Enabling the password policy.</li> <li>Oracle Directory Server Enterprise Edition (Destination Directory) pass-through authentication</li> <li>Synchronization using password hashing techniques</li> </ul>
Oracle Internet Directory	Microsoft Active Directory	<ul style="list-style-type: none"> <li>Enabling the password policy and reversible password encryption.</li> </ul>
Oracle Directory Server Enterprise Edition	Oracle Unified Directory	<ul style="list-style-type: none"> <li>Oracle Unified Directory (Destination Directory) pass-through authentication</li> <li>On-Demand Password</li> <li>Password Translate</li> <li>Synchronization using password hashing techniques</li> </ul>
Oracle Directory Server Enterprise Edition	Oracle Internet Directory	<ul style="list-style-type: none"> <li>Oracle Internet Directory External Authentication Plug-ins</li> <li>Password Translate</li> <li>Synchronization using password hashing techniques</li> </ul>
Oracle Directory Server Enterprise Edition	Oracle Directory Server Enterprise Edition	<ul style="list-style-type: none"> <li>Oracle Directory Server Enterprise Edition (destination directory) pass-through authentication</li> <li>On-Demand Password</li> <li>Password Translate</li> <li>Synchronization using password hashing techniques</li> </ul>
Oracle Directory Server Enterprise Edition	Microsoft Active Directory	Password Translate
Microsoft Active Directory	Oracle Unified Directory	<ul style="list-style-type: none"> <li>On-Demand Password</li> <li>Oracle Password Filter</li> </ul>
Microsoft Active Directory	Oracle Internet Directory	<ul style="list-style-type: none"> <li>Oracle Internet Directory External Authentication Plug-ins</li> <li>Oracle Password Filter</li> </ul>
Microsoft Active Directory	Oracle Directory Server Enterprise Edition	<ul style="list-style-type: none"> <li>On-Demand Password</li> <li>Oracle Password Filter</li> </ul>

## 9.8.2 Configuring Password Synchronization for Oracle Unified Directory

You can synchronize the password between the connected directory to the Oracle Unified Directory used as the back-end directory.

### Topics:

- [Synchronizing Password from a Connected Directory to the Oracle Unified Directory using On-Demand Password](#)
- [Synchronizing Password from the Oracle Unified Directory to a Connected Directory using Password Translate](#)

### Note:

Ensure that you configured Oracle Unified Directory for Oracle Directory Integration Platform using `dipConfigurator setup` command, as described in [Configuring the Oracle WebLogic Server Domain for Oracle Directory Integration Platform with Oracle Unified Directory](#).

### 9.8.2.1 Synchronizing Password from a Connected Directory to the Oracle Unified Directory using On-Demand Password

You can synchronize the password from a connected directory to Oracle Unified Directory by configuring an import profile. To do so, complete the following steps:

1. Configure the Oracle Directory Integration Platform plug-ins by running the `dipConfigurator setupPlugin` command on the command line and enter the following arguments:

**Table 9-7 setupPlugin Properties**

Argument	Definition
wlshost	Oracle WebLogic Server host name where Oracle Directory Integration Platform is deployed. The default value is localhost.
wlsport	Listening port number of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed. The default value is 7001.
wlsuser	Oracle WebLogic Server login ID.
ldaphost	Oracle Unified Directory host name. The default value is localhost.
ldapport	Oracle Unified Directory server port number. The default value is 636.
isldapssl	Enable or disable SSL. The default value is true.
ldapuser	The bind DN to connect to the directory. The default value is cn=Directory Manager.

**Table 9-7 (Cont.) setupPlugin Properties**

Argument	Definition
ldapadminport	The administration port number of the Oracle Unified Directory to which you want to connect. The default port number is 4444.

 **Note:**

You can view the `dipConfig.log` file, located at `<ORACLE_HOME>/ldap/log/`.

**Example:**

```
$ORACLE_HOME/bin/dipConfigurator setupPlugin -wlshost localhost -wlsport
7001 -wlsuser weblogic -ldaphost oudhost -ldapport 389 -ldapuser
"cn=Directory Manager" -isldapssl false -ldapadminport 4444
```

2. Create an attribute mapping rule using the Oracle Enterprise Manager Fusion Middleware Control:
  - a. Log in to the Oracle Enterprise Manager Fusion Middleware Control.
  - b. In the navigation panel on the left, click or expand **Identity and Access** and then select the DIP component where you want to edit the synchronization profile.
  - c. Click the **DIP Server** menu, point to **Administration**, and then select the synchronization profile you created for Oracle Unified Directory.
  - d. Click **Edit**. The Edit Synchronization Profile screen appears for the profile you want to edit.
  - e. Select the **Mapping** tab.
  - f. In the Attribute Mapping Rules section select the **Create** icon.  
The Add Attribute Mapping Rule screen is displayed.
  - g. Enter the following parameters:
    - **Source ObjectClass:** Select the required option. For example, select **User** if you are using Microsoft Active Directory as the connected directory.
    - **Attribute(s):** Select **pwdLastSet**.
    - **DIP-OUO ObjectClass:** Select **top**.
    - **DIP-OUO Attribute:** Select **orclodippwdlastset**.
    - **Mapping Expression:** Depending on the connected directory select an option. For example:
      - Microsoft Active Directory: `onDemandPassword(pwdLastSet)`
      - Oracle Unified Directory: `onDemandPassword(pwdChangedTime)`
      - Oracle Directory Server Enterprise Edition: `onDemandPassword(pwdChangedTime)`

 **Note:**

The `onDemandPassword` attribute is available, if the password expiration is configured for Oracle Directory Server Enterprise Edition. For more information, see "[Policy for Password Expiration](#)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.

Click **OK**.

You can click the **Validate All Mapping Rules** button to test your mapping rules.

 **Note:**

You can also edit the mapping rules by adding the following attribute mapping rule:

```
pwdLastSet : : : user : orclODIPPwdLastSet : : top :  
onDemandPassword(pwdLastSet)
```

For more information, see [Adding an Entry to the Mapping Rules File](#).

## 9.8.2.2 Synchronizing Password from the Oracle Unified Directory to a Connected Directory using Password Translate

You can synchronize the password from Oracle Unified Directory (back-end directory) to a connected directory by configuring an export profile. To do so, complete the following steps:

1. If the Oracle Directory Integration Platform plug-ins is not configured, then you must run the `dipConfigurator setupPlugin` command on the command line:

```
$ORACLE_HOME/bin/dipConfigurator setupPlugin -wlshost localhost -wlsport  
7001 -wlsuser weblogic -ldaphost oudhost -ldapport 389 -ldapuser  
"cn=Directory Manager" -isldapssl false -ldapadminport 4444
```

For more information on the arguments, see [Table 9-7](#).

 **Note:**

Do not run the `dipConfigurator setupPlugin` command if you have already executed the command in [Synchronizing Password from a Connected Directory to the Oracle Unified Directory using On-Demand Password](#).

2. Run `dipConfigurator setupPasswordTranslation` command on the command line:

```
$ORACLE_HOME/bin/dipConfigurator setupPasswordTranslation -wlshost localhost  
-wlsport 7001 -wlsuser weblogic -ldaphost oudhost -ldapport 389 -ldapuser  
"cn=Directory Manager" -isldapssl false -ldapadminport 4444
```

For more information on the arguments, see [Table 9-7](#).

 **Note:**

You can view the `dipConfig.log` file, located at `<ORACLE_HOME>/ldap/log/`.

3. Create an attribute mapping rule using the Oracle Enterprise Manager Fusion Middleware Control:
    - a. Log in to Oracle Enterprise Manager Fusion Middleware Control.
    - b. In the navigation panel on the left, click or expand **Identity and Access** and then select the DIP component where you want to edit the synchronization profile.
    - c. Click the **DIP Server** menu, point to **Administration**, and then select the synchronization profile you created for Oracle Unified Directory.
    - d. Click **Edit**. The **Edit Synchronization Profile** screen appears for the profile you want to edit.
    - e. Select the **Mapping** tab.
    - f. In the Attribute Mapping Rules section select the **Create** icon.  
The Add Attribute Mapping Rule screen is displayed.
    - g. Enter the following parameters:
      - **Source ObjectClass**: Select the required option. For example, select **User** if you are using Microsoft Active Directory as the connected directory.
      - **Attribute(s)**: Select **unicodepwd**.
      - **DIP-OLD ObjectClass**: Select **top**.
      - **DIP-OLD Attribute**: Select **orclodiptranslatepassword**.
      - **Mapping Expression**: Enter `orclODIPTranslatePassword`.
- Click **OK**.
- You can click the **Validate All Mapping Rules** button to test your mapping rules.

 **Note:**

You can also edit the mapping rules by adding the following attribute mapping rule:

```
orclodiptranslatepassword: : : : unicodepwd : : user :
passwordtranslate(orclodiptranslatepassword)
```

For more information, see [Adding an Entry to the Mapping Rules File](#).

## 9.8.3 Configuring Password Synchronization for Oracle Directory Server Enterprise Edition

This is an optional step, you can import or export the password from the connected directory to the Oracle Directory Server Enterprise Edition used as the back-end directory.

### Topics:

- [Synchronizing Password from a Connected Directory to the Oracle Directory Server Enterprise Edition using On-Demand Password](#)
- [Synchronizing Password from the Oracle Directory Server Enterprise Edition to a Connected Directory Using Password Translate](#)

### 9.8.3.1 Synchronizing Password from a Connected Directory to the Oracle Directory Server Enterprise Edition using On-Demand Password

You can synchronize the password from a connected directory to Oracle Directory Server Enterprise Edition by configuring an import profile. To do so, complete the following steps:

1. Configure the Oracle Directory Integration Platform plug-ins by running the `dipConfigurator setupPlugin (<ORACLE_HOME>/bin)` command on the command line and enter the following arguments:

**Table 9-8 setupPlugin Properties**

Argument	Definition
wlshost	Oracle WebLogic Server host name where Oracle Directory Integration Platform is deployed. The default value is localhost.
wlsport	Listening port number of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed. The default value is 7001.
wlsuser	Oracle WebLogic Server login ID.
ldaphost	Oracle Directory Server Enterprise Edition host name. The default value is localhost.
ldapport	Oracle Directory Server Enterprise Edition server port number. The default value is 636.

**Table 9-8 (Cont.) setupPlugin Properties**

Argument	Definition
isldapssl	Enable or disable SSL. The default value is true.
ldapuser	The bind DN to connect to the directory. The default value is cn=Directory Manager.
ldapadminport	The administration port number of the Oracle Unified Directory to which you want to connect. The default port number is 4444.

**Example:**

```
$ORACLE_HOME/bin/dipConfigurator setupPlugin -wlshost localhost -wlsport
7001 -wlsuser weblogic -ldaphost odseehost -ldapport 636 -ldapuser
"cn=Directory Manager" -isldapssl true
```

 **Note:**

You can view the dipConfig.log file, located at <ORACLE\_HOME>/ldap/log/.

2. Restart the directory server instance:
 

```
dsadm restart
```
3. Create an attribute mapping rule using the Oracle Enterprise Manager Fusion Middleware Control:
  - a. Log in to the Oracle Enterprise Manager Fusion Middleware Control.
  - b. In the navigation panel on the left, click or expand **Identity and Access** and then select the DIP component where you want to edit the synchronization profile.
  - c. Click the **DIP Server** menu, point to **Administration**, and then select the synchronization profile you created for Oracle Directory Server Enterprise Edition.
  - d. Click **Edit**. The Edit Synchronization Profile screen appears for the profile you want to edit.
  - e. Select the **Mapping** tab.
  - f. In the Attribute Mapping Rules section select the **Create** icon. The Add Attribute Mapping Rule screen is displayed.
  - g. Enter the following parameters:
    - **Source ObjectClass:** Select the required option. For example, select **User** if you using Microsoft Active Directory as the connected directory.
    - **Attribute(s):** Select **pwdLastSet**.
    - **DIP-ODSEE ObjectClass:** Select **operational attributes**.
    - **DIP-ODSEE Attribute:** Select **orclodippwdlastset**.
    - **Mapping Expression:** Enter `onDemandPassword(pwdChangedTime)` as the mapping expression.



Click **OK**.

You can click the **Validate All Mapping Rules** button to test your mapping rules.

4. Add the following attribute mapping rule to the synchronization profile:

```
pwdLastSet : : user : orclODIPPwdLastSet : : top :
onDemandPassword(pwdLastSet)
```

The `orclSourceObjectDN` attribute is needed by the plug-ins. It belongs to several `objectClasses`: `orclSunOneObject`, `orclADObject`, `orclNDSObject`, `orclOpenLDAPObject`, and `orclTDSObject`. A rule assigning this value must be included in the (import) profile, although the templates already include it. For example:

```
targetdn : :top:orclSourceObjectDN: :orclADObject:
```

For more information, see [Adding an Entry to the Mapping Rules File](#).

### 9.8.3.2 Synchronizing Password from the Oracle Directory Server Enterprise Edition to a Connected Directory Using Password Translate

You can synchronize the password from Oracle Directory Server Enterprise Edition (back-end directory) to a connected directory by configuring an export profile. To do so, complete the following steps:

1. Configure the Oracle Directory Integration Platform plug-ins by running the `dipConfigurator setupPlugin (<ORACLE_HOME>/bin)` command on the command line:

```
$ORACLE_HOME/bin/dipConfigurator setupPlugin -wlshost localhost -wlsport
7001 -wlsuser weblogic -ldaphost odseehost -ldapport 636 -ldapuser
"cn=Directory Manager"
```

For more information on the arguments, see [Table 9-8](#).

2. Restart the directory server instance:

```
dsadm restart
```

3. Run `dipConfigurator setupPasswordTranslation (<ORACLE_HOME>/bin)` command on the command line:

```
$ORACLE_HOME/bin/dipConfigurator setupPasswordTranslation -wlshost localhost
-wlsport 7001 -wlsuser weblogic -ldaphost odseelocalhost -ldapport 636 -
ldapuser "cn=Directory Manager" -isldapssl true
```

For more information on the arguments, see [Table 9-8](#).

#### Note:

You can view the `dipConfig.log` file, located at `<ORACLE_HOME>/ldap/log/`.

4. Edit the attribute mapping rule using the Oracle Enterprise Manager Fusion Middleware Control:

- a. Log in to Oracle Enterprise Manager Fusion Middleware Control.
- b. In the navigation panel on the left, click or expand **Identity and Access** and then select the DIP component where you want to edit the synchronization profile.
- c. Click the **DIP Server** menu, point to **Administration**, and then select the synchronization profile you created for Oracle Directory Server Enterprise Edition.
- d. Click **Edit**. The **Edit Synchronization Profile** screen appears for the profile you want to edit.
- e. Select the **Mapping** tab.
- f. In the Attribute Mapping Rules section select the **Create** icon.  
The Add Attribute Mapping Rule screen is displayed.
- g. Enter the following parameters:
  - **Source ObjectClass**: Select the required option. For example, select **User** if you using Microsoft Active Directory as the connected directory.
  - **Attribute(s)**: Select **unicodepwd**.
  - **DIP-ODSEE ObjectClass**: Select **operational attributes**.
  - **DIP-ODSEE Attribute**: Select **orclodiptransalepassword**.
  - **Mapping Expression**: Enter `orclODIPTranslatePassword` as the mapping expression.

Click **OK**.

You can click the **Validate All Mapping Rules** button to test your mapping rules.

 **Note:**

You can also edit the mapping rules by adding the following attribute mapping rule:

```
orclodiptranslatepassword: : : top : unicodepwd : : user :  
passwordtranslate(orclodiptranslatepassword)
```

For more information, see [Adding an Entry to the Mapping Rules File](#).

## 9.8.4 Configuring Password Synchronization for Oracle Internet Directory

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.

For example, IBM Tivoli Directory Server and Oracle Directory Server Enterprise Edition 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:**

Oracle Internet Directory supports multiple password policies in each realm, commonly known as Fine-Grained Password Policies.

See, 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 `orclPwPolicyEnable` attribute in the appropriate container. To enable reversible password encryption in the Oracle Internet Directory server, assign a value of 1 to the `orclPwEncryptionEnable` 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 `orclPwPolicyEnable` and `orclPwEncryptionEnable` attributes in the following entry:

```
cn=default,cn=PwPolicyEntry,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=PwPolicyEntry,cn=common,cn=products,cn=oraclecontext,Realm_DN
changetype: modify
replace: orclPwPolicyEnable
orclPwPolicyEnable: 1
-
replace: orclPwEncryptionEnable
orclPwEncryptionEnable: 1
```

 **See Also:**

*Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory* for information on managing Oracle Internet Directory password policies.

# 10

## Managing Directory Synchronization Profiles

You can manage directory synchronization profiles using Fusion Middleware Control or the `manageSyncProfiles` utility.

### Topics:

- [Managing Synchronization Profiles Using Fusion Middleware Control](#)
- [Managing Synchronization Profiles Using `manageSyncProfiles`](#)
- [Modifying the Synchronization Status Attributes](#)
- [Setting Null Values in Synchronization Profiles](#)
- [manageSyncProfiles utility](#)

### 10.1 Managing Synchronization Profiles Using Fusion Middleware Control

You can create, modify, and delete synchronization profiles using Oracle Enterprise Manager Fusion Middleware Control.

### 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.

#### 10.1.1 Creating Synchronization Profiles

This topic 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, select **Administration**, and then click **Synchronization Profiles**.

The **Manage Synchronization Profiles** page 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.

#### Example 10-1 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 pushes the changes from the Oracle back-end directory into the connected directory. Selecting Destination pushes changes from the connected directory into the Oracle back-end 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.

 **Note:**

Ensure that the connected directories cipher suites are supported in Oracle Directory Integration Platform and available out-of-box. See [Supported Out-of-Box Cipher Suites](#).

- **Database Service Name:** This option appears, if you have selected "Database (JDBC)" from the Type menu. Enter the database service name.

 **Note:**

Enter the service name, not the SID.

- **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.

### Example 10-2 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 you have errors in the mapping rules that you have defined, then the profile will not work. However, warnings do not prevent the profile from being used.

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

- **DIP-OID Container / DIP-OU 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 are 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 are 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-OU 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.

 **Note:**

After registering the profile, the **Source Attribute Required** option is enabled by default. Ensure that you disable the source attribute requirement to start synchronization.

- **DIP-OID Object Class / DIP-OUO 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-OUO 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-OUO 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.

### Example 10-3 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**.

**Example 10-4 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:** The maximum number of times the synchronization is to be retried during each synchronization cycle iteration. The interval of each synchronization cycle iteration is every 60 seconds, if the Scheduling Interval parameter is set to 60 seconds.
- **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.

 **Note:**

- \* 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.
- **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.

## 10.1.2 Editing Synchronization Profiles

You can edit an existing synchronization profile using Oracle Enterprise Manager Fusion Middleware Control.

To do so, perform 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/lem`.
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](#)" 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.

## 10.1.3 Enabling and Disabling Synchronization Profiles

You can enable or disable an existing synchronization profile using Oracle Enterprise Manager Fusion Middleware Control.

To do so, complete the following:

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/lem`.

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 disable the profile.

## 10.1.4 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.

## 10.1.5 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.

- [Running DIP Tester From the Enterprise Manager User Interface](#)
- [Running DIP Tester From the WLST Command-Line Interface](#)

## 10.1.5.1 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.

### Example 10-5 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 10-1 Synchronization Profile Properties, Basic Properties**

Property	Description
Profile Name	The name of the profile to be tested.
Synchronization Mode	Indicates the direction of synchronization. <i>Import</i> propagates changes from a connected directory to the Oracle back-end directory. <i>Export</i> 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 10-2 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.

**Table 10-2 (Cont.) Synchronization Profile Properties, Source Details or Destination Details**

Property	Description
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 10-3 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	<p>If true, specifies that the profile should continue the synchronization job if an error occurs. If false, the sync job is aborted.</p> <p>To change this value, edit the synchronization profile, select the <b>Advanced</b> tab, and edit the parameter in the <b>Additional Configuration Parameters</b> section.</p>
Update Search Count	<p>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.</p> <p>To change this value, edit the synchronization profile, select the Advanced tab, and edit the parameter in the <b>Additional Configuration Parameters</b> section.</p>

**Example 10-6 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.

**Example 10-7 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 from 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.



- **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.

### Example 10-8 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.

### Example 10-9 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.

### 10.1.5.2 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 | -ldiff 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>]
```

## 10.2 Managing Synchronization Profiles Using `manageSyncProfiles`

You can use the `manageSyncProfiles` utility to create and manage synchronization profiles from a command line.

For more information, see [manageSyncProfiles Utility](#).

## 10.3 Modifying the Synchronization Status Attributes

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

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.

To change the `orcllastappliedchangenumber` attribute using Oracle Enterprise Manager Fusion Middleware Control, perform the steps in "[Editing Synchronization Profiles](#)", 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`](#)".

## 10.4 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.

## 10.5 `manageSyncProfiles` Utility

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.

### Note:

- 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.

## 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]
```

## 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:**

[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 [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.

 **Note:**

Ensure that the connected directories cipher suites are supported in Oracle Directory Integration Platform and available out-of-box. See [Supported Out-of-Box Cipher Suites](#).

`-prfSt | -profileStatus`

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

`-help`

Provides command usage help.

### 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
```

# 11

## 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.

### Topics:

- [Overview of Directory Bootstrapping Using syncProfileBootstrap](#)
- [Bootstrapping in SSL Mode](#)
- [syncProfileBootstrap Utility](#)

### 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*.

## 11.1 Overview of Directory Bootstrapping Using syncProfileBootstrap

Use the `syncProfileBootstrap` utility to bootstrap between a connected directory and the Oracle back-end directory.

For more information, see [syncProfileBootstrap Utility](#).

### Note:

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 [Configure the Additional Configuration Information File](#) for more information.

### Topics:

- [Recommended Bootstrapping Methodology](#)

- [About How to Use a Parameter File to Bootstrap](#)
- [Bootstrapping from an LDAP File to an LDIF File](#)
- [Bootstrapping Using an LDIF File](#)
- [Bootstrapping Directly Using the Default Integration Profile](#)

### 11.1.1 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.

### 11.1.2 About How to Use a Parameter File to Bootstrap

You can use the parameter file to bootstrap between a connected directory and the Oracle back-end directory.

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.

### 11.1.3 Bootstrapping from an LDAP File to an LDIF File

This section describes how to bootstrap from an LDAP file to 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/conf`. 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. Examine the diagnostic log file for any error at the following location:

```
(UNIX) DOMAIN_HOME/servers/server_name/logs
(Windows) DOMAIN_HOME\servers\server_name\logs
```

The default name of the log file is `server-name-diagnostic.log`.

## 11.1.4 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](#)

### 11.1.4.1 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 -h HOST -p PORT -D wlsuser -profile
profile_name -loadParallelism threads -loadRetry retries
```

For more information, see [syncProfileBootstrap Utility](#).

5. 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
```

For more information, see [manageSyncProfiles Utility](#).

### 11.1.4.2 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/conf`.
5. Examine the diagnostic log file for any error at the following location:

```
(UNIX) DOMAIN_HOME/servers/server_name/logs
(Windows) DOMAIN_HOME\servers\server_name\logs
```

The default name of the log file is `server-name-diagnostic.log`.

6. To continue synchronization and update the last change number, use the `manageSyncProfiles updatechgnum` command:

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

## 11.1.5 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 `server-name-diagnostic.log` file located in the `DOMAIN_HOME/servers/server_name/logs` directory.

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

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"](#).
2. Enter the following command:

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

3. Examine the diagnostic log file for any error at the following location:

```
(UNIX) DOMAIN_HOME/servers/server_name/logs  
(Windows) DOMAIN_HOME\servers\server_name\logs
```

The default name of the log file is `server-name-diagnostic.log`.

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

## 11.2 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.



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.

 **Note:**

Oracle Directory Integration Platform 12c supports Transport Layer Security (TLS) v1.2 protocol for communication between a connected directory and the Oracle back-end directory. See [Transport Layer Security Protocol and Cipher Suites](#).

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. Invoke the WLST command:

```
(UNIX) ORACLE_HOME/oracle_common/common/bin/wlst.sh
(Windows) ORACLE_HOME\oracle_common\common\bin\wlst.cmd
```

- 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")
```

 **Note:**

Oracle Directory Integration Platform provides you the option to add cipher suites used by the connected directories, which is not available out-of-box. See [Adding Cipher Suites into Oracle Directory Integration Platform](#)

## 11.3 syncProfileBootstrap Utility

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

 **Note:**

- 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 [Configure 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.

### 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]
```

### Arguments for syncProfileBootstrap

The following table describes the arguments for `syncProfileBootstrap` utility.

**Table 11-1 syncProfileBootstrap utility Arguments**

Argument	Description
<b>-h   -host</b>	Oracle WebLogic Server where Oracle Directory Integration Platform is deployed.
<b>-p   -port</b>	Listening port of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed.
<b>-D   -wlsuser</b>	Oracle WebLogic Server login ID. <b>Note:</b> 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 <code>dipStatus</code> 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.
<b>-f   -file</b>	Bootstrap properties file.
<b>-pf   -profile</b>	The name of the synchronization profile to use when performing the operation.
<b>-ssl</b>	Executes the command in SSL mode. <b>Note:</b> The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. For more information, see "Configuring SSL" in <i>Oracle Fusion Middleware Securing Oracle WebLogic Server</i> .
<b>-pwd   -pwdOverNonSSL</b>	Forces the synchronization of the <code>userPassword</code> attribute over Non-SSL mode.
<b>-keystorePath</b>	The full path to the keystore.
<b>-keystoreType</b>	The type of the keystore identified by <code>-keystorePath</code> . For example: <code>-keystorePath jks</code> or <code>-keystorePath PKCS12</code>
<b>-lp   -loadParallelism</b>	Indicates that loading to the Oracle back-end directory is to take place in parallel by using multiple threads. For example, <code>-loadparallelism 5</code> means that 5 threads are to be created, each of which tries to load the entries in parallel to the Oracle back-end directory.
<b>-lr   -loadRetry</b>	The number of times the retry should be made (when the load to the destination fails) before marking the entry as bad entry.
<b>-help</b>	Provides usage help for the command.

**Tasks and Examples for syncProfileBootstrap**

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

```
syncProfileBootstrap -h myhost.mycompany.com -p 7005 -D login_ID \  
-f /opt/ldap/odip/bootstrap.properties -lr 3
```

```
syncProfileBootstrap -h myhost.mycompany.com -p 7005 -D weblogic -fp  
ImportProfile -pwd
```

# 12

## Synchronizing with Tables in Oracle Database

Know more about the `DBReader` configuration and `dbexport` configuration files to synchronize data stored in Oracle Database tables with the Oracle back-end directory and an Oracle back-end directory with an Oracle Database.

### Note:

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.

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 [Managing Synchronization Profiles Using Fusion Middleware Control](#), for help creating and managing profiles using the user interface [Managing Synchronization Profiles Using `manageSyncProfiles`](#), for help creating and managing profiles using the WLST `manageSyncProfiles` command utility.

### Topics:

- [Overview of the Additional Configuration Information File](#)
- [Updating the Configuration File](#)
- [Preparing the Directory Integration Profile](#)
- [Example: Synchronizing a Relational Database Table to the Back-end Directory](#)



**Note:**

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

- [Introduction to Oracle Directory Integration Platform](#)
- [Understanding the Oracle Directory Synchronization Service](#)

## 12.1 Overview of the Additional Configuration Information File

Using `DBReader` configuration files, 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.

Modify the `dbexport` configuration files, if you want to create from the command line an export profile that will synchronize an Oracle back-end directory with an Oracle Database.

- [Preparing the Additional Configuration Information File](#)
- [Understanding the Format of the Additional Configuration Information File](#)
- [Understanding the `DBReader.cfg.master` Configuration File](#)
- [Understanding the `dbexport.cfg.master` Configuration File](#)

### 12.1.1 Preparing the Additional Configuration Information File

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

### 12.1.2 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

```

### 12.1.3 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,'@example.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, '@example.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.

## 12.1.4 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.

## 12.2 Updating the Configuration File

You must configure the `DBReader.map.master` and `dbexport.map.master` configuration files.

**Topics:**

- [The DBReader.map.master Configuration File](#)
- [The dbexport.map.master Configuration File](#)

### 12.2.1 The DBReader.map.master Configuration File

You can update the `DBReader.map.master` configuration file to configure mapping rules.

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.

## 12.2.2 The `dbexport.map.master` Configuration File

You can configure the `dbexport.map.master` configuration File to configure mapping rules.

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.

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.

## 12.3 Preparing the Directory Integration Profile

You can create a directory integration profile by using the Oracle Enterprise Manager Fusion Middleware Control user interface.

See [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 [Managing Synchronization Profiles Using manageSyncProfiles](#).

To configure the directory integration profile, follow the general instructions in "[Registering Connectors in Oracle Directory Integration Platform](#)", 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.

## 12.4 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:

- [About the Example Scenario](#)
- [Configure the Additional Configuration Information File](#)
- [Configure the Mapping File](#)
- [Configure the Directory Integration Profile](#)
- [Uploading the Additional Configuration Information and Mapping Files](#)
- [Understanding Synchronization Process for Relational Database Table with the Oracle Back-End Directory](#)
- [Observations About the Synchronizing a Relational Database Table to the Back-End Directory Example](#)

### 12.4.1 About the Example Scenario

This example shows how 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@EXAMPLE.COM	435-324-3455
98360	ROGER BECK	3-JUL-2001	ROGER.BECK@EXAMPLE.COM	435-324-3600
98365	JIMMY WONG	4-MAR-2001	JIMMY.WONG@EXAMPLE.COM	435-324-2390
98370	GEORGE TWINSLEY	6-FEB-2002	GEORGE.TWINSLEY@EXAMPLE.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.example.com`, the database listener port is `1526`, and the `serviceName` is `ias.example.com`. The database URL is `machine.example.com:1526:ias.example.com`.
- 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.

## 12.4.2 Configure the Additional Configuration Information File

This example uses the same Additional Configuration Information file described earlier in "[Overview of the Additional Configuration Information File](#)".

## 12.4.3 Configure 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.

## 12.4.4 Configure the Directory Integration Profile

The directory integration profile for this example contains the attribute values as described in [Table 12-1](#). 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”](#).

**Table 12-1 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.agentexeccommand)	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
Connected Directory Account Password (odip.profile.condirpassword)	testdbsyncpwd
Connected Directory URL (odip.profile.condirurl)	machine.example.com:1526:iasdb.example.com Where <code>iasdb.example.com</code> is the database service name.
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 <code>employeenumber</code> 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 <code>orclodipOIDMatchingFilter</code> attribute is unique in the database also. Once a database row is retrieved, the Oracle Directory Integration Platform searches the directory for that <code>employeenumber</code> in the domain <code>dc=testdbsync,dc=com</code> 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.

**Table 12-1 (Cont.) Directory Integration Profile for TESTDBIMPORT**

Attribute	Value
Last Applied Change Number ( <code>odip.profile.lastchgnum</code> )	20000101000000  This means that the first time the profile executes, it retrieves and synchronizes all four rows. Subsequently, it retrieves rows only when the <code>LAST_UPDATE_DATE</code> column in the table is updated to the time last modified.

## 12.4.5 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.

Run the `manageSyncProfiles` command, as follows:

```
manageSyncProfiles update -h HOST -p PORT -D WLS_USER -pf PROFILE_NAME -file
FILE_NAME
```

## 12.4.6 Understanding Synchronization Process for Relational Database Table with the Oracle Back-End Directory

This section explains the synchronization process for relational database table with the Oracle back-end directory.

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.

## 12.4.7 Observations About the Synchronizing a Relational Database Table to the Back-End Directory Example

This topic describes the form of the row after it is retrieved from the database.

```
EmpNum: 98357
EName: JOHN DOE
UID: JOHN.DOE
EMAIL: JOHN.DOE@EXAMPLE.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@EXAMPLE.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`)

# 13

## 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.

### Topics:

- [Introduction to Synchronization with Oracle Human Resources](#)
- [What are the Data you can Import from Oracle Human Resources?](#)
- [Managing Synchronization Between Oracle Human Resources and the Oracle Back-end Directory](#)
- [Understanding the Synchronization Process between Oracle Human Resources and the Oracle Back-End directory](#)
- [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 [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.

## 13.1 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.

## 13.2 What are the Data you can Import from Oracle Human Resources?

This section describes the attributes that you can import from Oracle Human Resources.

[Table 13-1](#) lists the tables in the Oracle Human Resources schema. If you choose, you can import most of these attributes into Oracle Internet Directory.

**Table 13-1 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 13-2](#) shows some of the fields in the Oracle Human Resources user interface. These fields appear when you add or modify employee data.

**Table 13-2 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



**Table 13-2 (Cont.) Fields in the Oracle Human Resources User Interface**

ATTRIBUTE NAME	DESCRIPTION	FORM/CANVAS/FIELD_NAME
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
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

## 13.3 Managing Synchronization Between Oracle Human Resources and the Oracle Back-end Directory

Use the steps in the following order to manage synchronization between Oracle Human Resources and the Oracle Back-end Directory.

**Topics:**

- [Configure a Directory Integration Profile for the Oracle Human Resources Connector](#)
- [About Oracle Human Resources Attributes to be Synchronized with the Oracle Back-end Directory](#)
- [About Mapping Rules for the Oracle Human Resources Connector](#)
- [Setting Up Synchronization from Oracle Human Resources to the Oracle Back-end Directory](#)

## 13.3.1 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 [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 13-3](#).

**Table 13-3 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)	<p>The direction of synchronization between the Oracle back-end directory and a connected directory.</p> <ul style="list-style-type: none"> <li>• IMPORT indicates importing changes from a connected directory to the Oracle back-end directory.</li> <li>• EXPORT indicates exporting changes from the Oracle back-end directory to a connected directory.</li> </ul> <p>The default is IMPORT.</p> <p>This attribute is mandatory and modifiable.</p> <p><b>Note:</b> In Oracle Internet Directory 11g Release 1 (11.1.1), only import operations for Oracle Human Resources are supported.</p>
<b>Execution Information</b>	
Agent Execution Command (odip.profile.agentexeccommand)	<p>Connector executable name and argument list used by the directory integration server to execute the connector.</p> <p>This attribute is mandatory and modifiable.</p> <p>The default is:</p> <pre>odihragent OracleHRAgent connect=hrdb \ login=%odip.profile.condiraccount \ pass=%odip.profile.condirpassword \ date=%orclODIPLastSuccessfulExecutionTime \</pre> <p>You must set the value in the argument <code>connect=hrdb</code> to the connect string of the Oracle Human Resources system database.</p>
Connected Directory Account (odip.profile.condiraccount)	<p>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.</p> <p><b>See Also:</b> <a href="#">Synchronizing with Oracle Human Resources</a> for typical usage of passing it in the command-line</p>

**Table 13-3 (Cont.) Attributes Specific to Oracle Human Resources Connector Integration Profile**

Attribute	Description
Advanced Configuration Information (odip.profile.configfile)	<p>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.</p> <p>The value stored in this attribute represents (for Oracle Human Resources connector) all attributes that need to be synchronized from Oracle Human Resources.</p> <p><b>See Also:</b> <a href="#">"About Oracle Human Resources Attributes to be Synchronized with the Oracle Back-end Directory"</a></p> <p>This attribute is mandatory for the Oracle Human Resources connector.</p>
Connected Directory URL (odip.profile.condirurl)	<p>The host and port details of the connected directory. They must be entered in this format: <i>host:port:sid</i>.</p>
Interface Type (odip.profile.interface)	<p>The interface used for data transfer. Because it is in the form of a tagged file, it is set to TAGGED.</p> <p><b>Note:</b> You should not modify this attribute for Oracle Human Resources profile.</p>
<b>Mapping Information</b>	
Mapping Rules (odip.profile.mapfile)	<p>Attribute for storing the mapping rules. Store the mapping rules in a file by using Oracle Enterprise Manager Fusion Middleware Control. See <a href="#">Managing Directory Synchronization Profiles</a> for information on using Oracle Enterprise Manager Fusion Middleware Control.</p> <p>This attribute is mandatory for Oracle Human Resources and is modifiable.</p> <p><b>See Also:</b></p> <ul style="list-style-type: none"> <li>• <a href="#">"#unique_162/unique_162_Connect_42_1122151"</a></li> <li>• <a href="#">"Configuring Mapping Rules"</a></li> </ul>
Connected Directory Matching Filter (odip.profile.condirfilter)	<p>This is not used in Oracle Human Resources connectivity.</p>
OID Matching Filter (odip.profile.oidfilter)	<p>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.</p> <p>It is of the form <code>employeenumber=%</code>.</p> <p>It is optional and modifiable.</p>
<b>Status Information</b>	
OID Last Applied Change Number (odip.profile.lastchgnum)	<p>This attribute, standard for all export profiles, does not apply to Oracle Human Resources synchronization.</p>

## 13.3.2 About Oracle Human Resources 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.

- [Configure the List of Attributes to be Synchronized with the Oracle Back-end Directory](#)
- [Modifying Additional Oracle Human Resources Attributes for Synchronization](#)
- [Excluding Oracle Human Resources Attributes from Synchronization](#)
- [Configure a SQL SELECT Statement in the Configuration File to Support Complex Selection Criteria](#)

### 13.3.2.1 Configure the List of Attributes to be Synchronized with the Oracle Back-end Directory

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 13-4 describes columns in the default list of Oracle Human Resources attributes.

**Table 13-4 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.
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

```

### 13.3.2.2 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 13-1](#). 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.

### 13.3.2.3 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 `oraclehrgent.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.

### 13.3.2.4 Configure 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 `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')

```

### 13.3.3 About 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:

"[#unique\\_162/unique\\_162\\_Connect\\_42\\_I122151](#)" for the contents of the `oraclehragent.map.master` file and a description of the format of the mapping rules records

### 13.3.4 Setting Up Synchronization from Oracle Human Resources to the Oracle Back-end Directory

You must set up synchronization from Oracle Human Resources to the Oracle back-end directory.

To do so, perform the following steps:

1. Set the `ORACLE_HOME` environment variable in the environment where the `wls_ods1` managed server is running.
2. 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.
3. 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
4. Configure an integration profile for the Oracle Human Resources connector, as described in "[Configure a Directory Integration Profile for the Oracle Human Resources Connector](#)". 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
5. Once everything is properly set, set the Profile Status attribute to `ENABLE`. This indicates that the Oracle Human Resources connector is ready to run.
6. Start the Oracle directory server and the Oracle Human Resources system if they are not already running on the respective hosts.
7. When everything is ready, start the directory integration server if it is not already running on this host.

## 13.4 Understanding the Synchronization Process between Oracle Human Resources and the Oracle Back-End directory

This section explains the synchronization process between Oracle Human Resources and the Oracle back-end directory.

Once the Oracle Human Resources system, Oracle back-end 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 back-end 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.

## 13.5 Bootstrapping the Oracle Back-end Directory from Oracle Human Resources

You can use the Oracle Human Resources connector or external tools to bootstrap Oracle back-end directory from Oracle Human Resources.

To bootstrap Oracle back-end 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.

# 14

## Synchronizing with 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.

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.

### Topics:

- [About Change Logs](#)
- [Enabling Third-Party Metadirectory Solutions to Synchronize with the Oracle Back-end Directory](#)
- [Understanding Synchronization Process](#)
- [Disabling and Deleting Change Subscription Objects](#)

### 14.1 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](#)" for a conceptual discussion of directory integration profiles

## 14.2 Enabling Third-Party Metadirectory Solutions to Synchronize with the Oracle Back-end Directory

Perform the tasks described in this section to enable third-party metadirectory solutions to retrieve changes from the Oracle back-end directory.

- [Performing Initial Bootstrapping](#)
- [Understand How to Change Subscription Object in the Oracle Back-end Directory for the Third-Party Metadirectory Solution](#)

### 14.2.1 Performing Initial Bootstrapping

This section explains how to bootstrap a directory to synchronize data between a local directory and the Oracle back-end directory.

To do so, complete the following steps:

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*.

## 14.2.2 Understand How to 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.

### 14.2.2.1 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.

### 14.2.2.2 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](#)" for instructions about temporarily disabling or deleting change subscription objects

## 14.3 Understanding Synchronization Process

Learn more about the synchronization process with supported third-party metadirectory solutions.

**Topics:**

- [Retrieving Changes the First Time from the Oracle Back-end Directory](#)
- [Updating the `orclLastAppliedChangeNumber` Attribute in the Oracle Back-end Directory](#)

### 14.3.1 Retrieving Changes the First Time from the Oracle Back-end Directory

The following example illustrates how 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 "[Understand How to Change](#)

Subscription Object in the Oracle Back-end Directory for the Third-Party Metadirectory Solution".

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.

## 14.3.2 Updating 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.

## 14.4 Disabling and Deleting Change Subscription Objects

You can temporarily disable or delete an existing change subscription object.

### Topics:

- [Disabling a Change Subscription Object](#)
- [Deleting a Change Subscription Object](#)

## 14.4.1 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
```

## 14.4.2 Deleting a Change Subscription Object

Run the `ldapdelete` command to delete a change subscription object.

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 V

## 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.

- [Understanding the Oracle Directory Integration Platform for Provisioning](#)
- [Deploying Provisioning-Integrated Applications](#)
- [Understanding the Oracle Provisioning Event Engine](#)
- [Integration of Provisioning Data with Oracle E-Business Suite](#)



# 15

## Understanding the Oracle Directory Integration Platform for Provisioning

Know more about the Oracle Directory Integration Platform Provisioning.

### Topics:

- [What Is Provisioning?](#)
- [Oracle Directory Integration Platform Service Components](#)
- [Understanding Provisioning Concepts](#)
- [Overview of Provisioning Methodologies](#)
- [About Organization of User Profiles in the Oracle Back-End Directory](#)
- [Understanding Provisioning Flow](#)
- [Understanding the Delegation of Administrative Privileges](#)

### See Also:

Developing Provisioning-Integrated Applications in *Oracle Fusion Middleware Application Developer's Guide for Oracle Identity Management*.

### 15.1 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 Unified Directory or 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 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 Unified Directory or 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 back-end directory. User accounts can be created in the Oracle back-end directory with the following tools or methods:

- Oracle Directory Services Manager

- Bulk provisioning using `provProfileBulkProv` command-line interface
- 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 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.

## 15.2 Understanding Oracle Provisioning Products

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 Unified Directory 11g Release 2 (11.1.2.3) and above or Oracle Internet Directory 11g Release 1 (11.1.1) as your back-end directory server to use the Oracle Directory Integration Platform provisioning functionality.
- If you are using Oracle Unified Directory as your back-end directory for Oracle Directory Integration Platform provisioning functionality, then you must install Oracle Directory Integration Platform 11g Release 1 (11.1.1.9.0) or a higher version. Earlier versions of Oracle Directory Integration Platform or versions upgraded/patched to 11g Release 1 (11.1.1.9.0) do not support provisioning functionality for Oracle Unified Directory.

## 15.3 Oracle Directory Integration Platform Service Components

The Oracle Directory Integration Platform Service consists of the following components:

- The Oracle Directory Integration Platform.

 **See Also:**

[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 `manageProvProfiles`.

 **See Also:**

"[About manageProvProfiles Command](#)" for information about `manageProvProfiles`.

## 15.4 Understanding Provisioning Concepts

This section explains how applications are provisioned with Oracle Directory Integration Platform Provisioning.

**Topics:**

- [About Synchronous Provisioning](#)
- [About Asynchronous Provisioning](#)
- [About Provisioning Data Flow](#)

## 15.4.1 About Synchronous Provisioning

A provisioning-integrated application can maintain user information in the back-end directory or a connected repository. Applications that maintain user information in the back-end directory can use the Data Access Java plug-in to create, modify, and delete user entries whenever the change occurs in back-end directory.



### See Also:

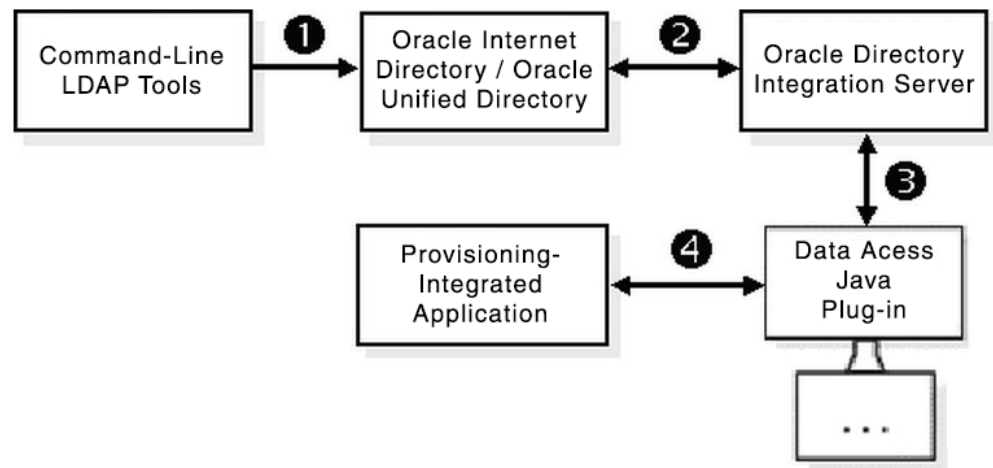
Section "Data Access Provisioning Plug-in" in the *Oracle Identity Management*.

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 the back-end 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 Back-end Directory](#)" for a list of provisioning status.

Synchronous provisioning follows this process:

1. A new user entry is created in the back-end directory from one of the following sources:
  - The Oracle Enterprise Manager user interface
  - Bulk provisioning using `provProfileBulkProv` command
  - 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 15-1](#) illustrates the process of how an application is synchronously provisioned using command-line LDAP tools.

**Figure 15-1 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 back-end directory.
2. At the next scheduled synchronization interval, the Oracle Directory Integration Platform identifies new user entries in the back-end 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.

## 15.4.2 About 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 the back-end 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 Back-end Directory](#)" for a list of provisioning status.

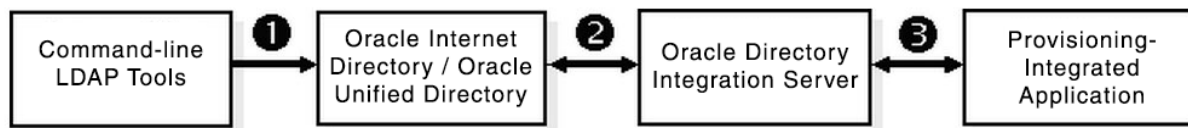
Asynchronous provisioning follows this process:

1. A new user entry and an associated entry containing application-specific user preferences are created in back-end directory from one of the following sources:
  - Oracle Enterprise Manager user interface
  - Bulk provisioning using 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 back-end directory that require provisioning.
  3. Provisioning events are sent from the Oracle Directory Integration Platform to the PL/SQL plug-in.

Figure 15-2 illustrates the process of how an application is asynchronously provisioned using command-line LDAP tools.

**Figure 15-2 Asynchronous Provisioning using Command-Line LDAP Tools**



As illustrated in Figure 15-2, asynchronous provisioning using command-line LDAP tools follows this process:

1. A new user entry is created in the Oracle Unified Directory or 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 back-end 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.

### 15.4.3 About Provisioning Data Flow

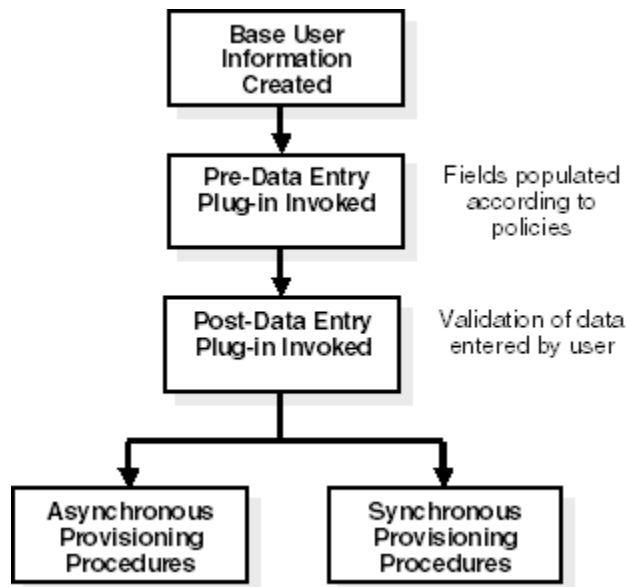
Regardless of whether it is provisioned synchronously or asynchronously, an application can invoke the Pre-Data Entry and Post-Data Entry plug-ins to enhance provisioning intelligence and implement business policies. Both plug-ins are invoked by bulk provisioning using the `provProfileBulkProv` command.

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 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 15-3 illustrates the provisioning data flow using the Pre-Data Entry and Post-Data Entry plug-ins.

**Figure 15-3 Provisioning Data Flow**



As illustrated in [Figure 15-3](#), 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.

## 15.5 Overview of Provisioning Methodologies

This section describes the procedures for provisioning users in Oracle Identity Management.

### Topics:

- [Understanding Provisioning Users that are Synchronized from an External Source](#)
- [Understanding Provisioning Users Created with Command-Line LDAP Tools](#)
- [Bulk Provisioning Using the provProfileBulkProv Tool](#)
- [On-Demand Provisioning](#)
- [Application Bootstrapping](#)

### 15.5.1 Understanding Provisioning Users that are Synchronized from an External Source

When Oracle Unified Directory or Oracle Internet Directory is used as a central repository, and enterprise user entries are synchronized from connected directories to the back-end directory, each user identity is automatically provisioned according to the default provisioning policy of each provisioning-integrated application.

## 15.5.2 Understanding 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 the back-end 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.

## 15.5.3 Bulk Provisioning Using the provProfileBulkProv Tool

Run the `provProfileBulkProv` utility, located in the `ORACLE_HOME/bin` directory, to perform initial migration of data from an LDIF file to the back-end directory for a provisioning profile.

For more information, see [provProfileBulkProv utility](#).

## 15.5.4 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 back-end directory.

## 15.5.5 Application Bootstrapping

The Oracle Directory Integration Platform Service notifies newly registered applications of all existing user entries in Oracle Unified Directory or Oracle Internet Directory and attempts to provision each existing user entry as if they were a new user in the application.

# 15.6 About Organization of User Profiles in the Oracle Back-End Directory

This section discusses the organization of user profiles in the Oracle Unified Directory or Oracle Internet Directory back-end directory.

**Topics:**

- [Understanding Organization of Provisioning Entries in the Directory Information Tree](#)
- [Understanding User Provisioning Status](#)

## 15.6.1 Understanding 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.

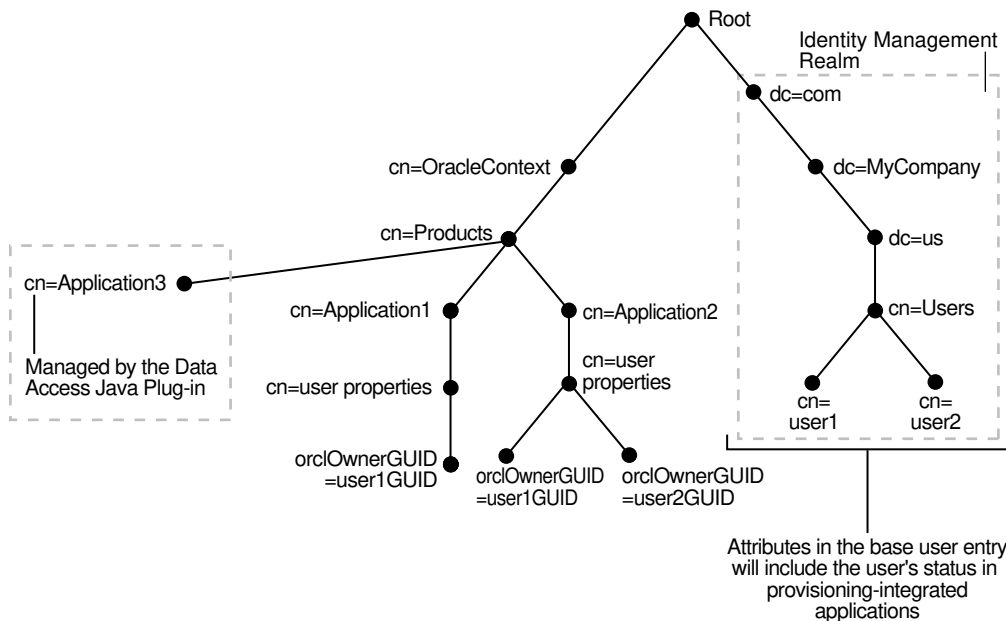
These user attributes for the Oracle Directory Integration Platform Service can be categorized as follows:

- 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` auxiliary class.

The 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 15-4](#), 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](#)". The Oracle Directory Integration Platform Service invokes this plug-in whenever the base user attributes or application-specific attributes are modified.

Figure 15-4 Oracle Back-End Directory Base User and Application-Specific Attributes



## 15.6.2 Understanding User Provisioning Status

This section discusses the user provisioning status in Oracle Unified Directory or Oracle Internet Directory.

### Topics:

- [Provisioning Status in Oracle Back-end Directory](#)
- [Provisioning Status Transitions](#)
- [Understanding Upgrade and Coexistence Provisioning Status](#)
- [Understanding Provisioning Status and Exception Handling](#)

### 15.6.2.1 Provisioning Status in Oracle Back-end Directory

The Oracle Provisioning Service records a user's provisioning status in the back-end 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 15-1](#) lists the provisioning status.

**Table 15-1 Provisioning Status in the Back-end Directory**

Internal Status	GUI Status	Description
<b>Provisioning Status</b>		
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.

**Table 15-1 (Cont.) Provisioning Status in the Back-end Directory**

Internal Status	GUI Status	Description
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.
<b>Deprovisioning Status</b>		
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.
<b>Upgrade Status</b>		
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 the back-end 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 status 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.

## 15.6.2.2 Provisioning Status Transitions

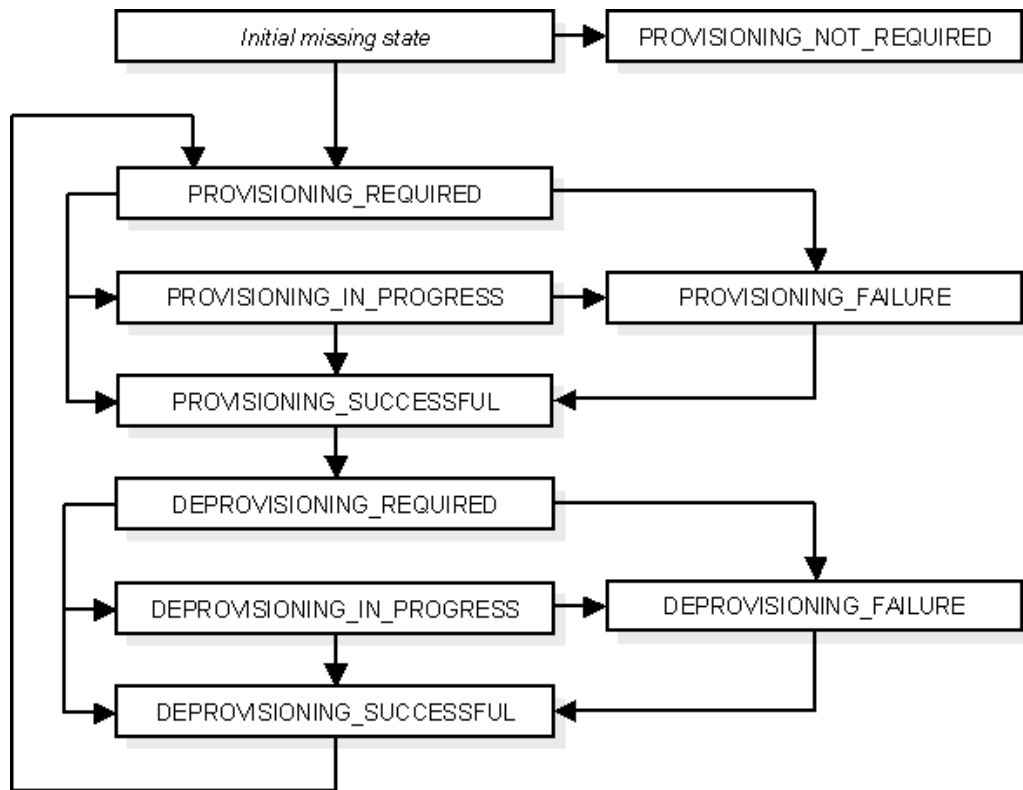
Table 15-2 lists the valid provisioning status transitions.

**Table 15-2 Valid Provisioning Status Transitions in the Back-end Directory**

Internal Status	GUI Status	Valid Transition From
<b>Provisioning Status</b>		
PROVISIONING_REQUIRED	Pending	<i>Initial missing state</i> DEPROVISIONING_SUCCESSFUL
PROVISIONING_IN_PROGRESS	In Progress	PROVISIONING_REQUIRED
PROVISIONING_SUCCESSFUL	Successful	PROVISIONING_REQUIRED PROVISIONING_IN_PROGRESS PROVISIONING_FAILURE
PROVISIONING_NOT_REQUIRED	Not Requested	<i>Initial missing state</i>
PROVISIONING_FAILURE	Failed	PROVISIONING_REQUIRED PROVISIONING_IN_PROGRESS
<b>Deprovisioning Status</b>		
DEPROVISIONING_REQUIRED	Pending de-provisioning	PROVISIONING_SUCCESSFUL
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 15-5 illustrates the valid provisioning status transitions.

**Figure 15-5 Valid Provisioning Status Transitions**



### 15.6.2.3 Understanding Upgrade and Coexistence Provisioning Status

In Oracle Identity Management, a user entry can be physically represented in the back-end 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 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.

## 15.6.2.4 Understanding Provisioning Status 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.

## 15.7 Understanding Provisioning Flow

This section discusses the flow of information and control in various provisioning scenarios.

### Topics:

- [Viewing and Editing Provisioning Profiles Using Fusion Middleware Control](#)
- [Provisioning User from an External Source](#)

### 15.7.1 Viewing and Editing Provisioning Profiles Using Fusion Middleware Control

You can view and edit provisioning profiles using the Oracle Enterprise Manager Fusion Middleware Control.

To do so, 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 15-3](#) lists and describes the provisioning profile fields:

**Table 15-3 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 or Application to OUD	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 or OUD 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 back-end directory.
Last Execution	Shows the status (Success/Failed) and execution time of the last provisioning attempt.

## 15.7.2 Provisioning User 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 back-end directory. Oracle Directory Integration Platform will provide ongoing synchronization between the back-end 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 the back-end 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.

## 15.8 Understanding the Delegation of Administrative Privileges

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:

### Note:

- Oracle Directory Integration Platform 11g Release 1 (11.1.1) interoperates with and supports Oracle Delegated Administration Services release 10.1.4.3.0 and higher for Oracle Internet Directory.
- Oracle Unified Directory does not support Oracle Delegated Administration Services.

### 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.



## 15.9 provProfileBulkProv utility

Use the `provProfileBulkProv` utility, located in the `ORACLE_HOME/bin` directory, to perform initial migration of data from an LDIF file to the back-end directory for a provisioning profile.

### Note:

- When you run `provProfileBulkProv` utility, ensure that you specify the absolute path. For example: `/tmp/bulk.ldif`.
- 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.

### 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]
```

### Arguments for provProfileBulkProv

The following table describes the arguments for `provProfileBulkProv` utility.

**Table 15-4** dipStatus utility Arguments

Argument	Description
<b>-h   -host</b>	Oracle WebLogic Server where Oracle Directory Integration Platform is deployed.
<b>-p   -port</b>	Listening port of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed.
<b>-D   -wlsuser</b>	Oracle WebLogic Server login ID. <b>Note:</b> 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 <code>dipStatus</code> 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.

Table 15-4 (Cont.) dipStatus utility Arguments

Argument	Description
<b>-f   -file</b>	LDIF file containing the data to be migrated.
<b>-realm</b>	The realm in which the users are to be provisioned.
<b>-ssl</b>	Executes the command in SSL mode. <b>Note:</b> The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. For more information, see "Configuring SSL" in <i>Oracle Fusion Middleware Securing Oracle WebLogic Server</i> .
<b>-keystorePath</b>	The full path to the keystore.
<b>-keystoreType</b>	The type of the keystore identified by <code>-keystorePath</code> . For example: <code>-keystorePath jks</code> or <code>-keystorePath PKCS12</code>
<b>-encoding</b>	Input file encoding.
<b>-help</b>	Provides usage help for the command.

#### 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
```

# 16

## Deploying Provisioning-Integrated Applications

This chapter explains how to deploy provisioning-integrated applications with the Oracle Provisioning Service.

### Topics:

- [Deploying Provisioning-Integrated Applications](#)
- [About manageProvProfiles Command](#)
- [Tasks and Examples for manageProvProfiles](#)
- [Registering Applications for Provisioning](#)
- [Understanding How to Configure Application Provisioning Properties](#)
- [manageProvProfiles Command](#)



### See Also:

[Managing the Oracle Directory Integration Platform](#)

## 16.1 Deploying Provisioning-Integrated Applications

Perform the steps described in this section to deploy provisioning-integrated applications with the Oracle Provisioning Service.

Complete the following steps:

1. Install the back-end directory and Oracle Directory Integration Platform.



### Note:

- Only Oracle Unified Directory and Oracle Internet Directory is supported for provisioning.
- If you are using Oracle Unified Directory for provisioning, ensure that Oracle Directory Integration Platform is enabled as described in [Installing Oracle Unified Directory](#).

2. Load user information into Oracle Internet Directory or Oracle Unified Directory.

 See Also:

- *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*
- *Oracle Fusion Middleware Administering Oracle Unified Directory*

3. Start the Oracle Directory Integration Platform.
4. Install the applications and use the `manageProvProfiles` to create a provisioning profile for each application. Refer to "[About manageProvProfiles Command](#)" for more information.
5. Configure application registration by following the procedures described in "[Registering Applications for Provisioning](#)".
6. Configure application provisioning by following the procedures described in "[Understanding How to Configure Application Provisioning Properties](#)".
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:

- "Monitoring Oracle Internet Directory" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.
- "Monitoring Oracle Unified Directory" in the *Oracle® Fusion Middleware Administering Oracle Unified Directory*.

## 16.2 About manageProvProfiles Command

Use the `manageProvProfiles` command to manage provisioning profiles.

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 `manageProvProfiles` command located in the `ORACLE_HOME/bin` directory. For more information, see [manageProvProfiles Command](#) and [Tasks and Examples for manageProvProfiles](#).

You can use the `manageProvProfiles` 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 `manageProvProfiles` 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 Unified Directory or 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 `manageProvProfiles` command unless prompted for one.

## 16.3 Tasks and Examples for manageProvProfiles

Use the `manageProvProfiles` to create, modify, delete, or disable a provisioning profile.

### Topics:

- [Creating a Provisioning Profile](#)
- [Modifying a Provisioning Profile](#)
- [Deleting a Provisioning Profile](#)
- [Disabling a Provisioning Profile](#)
- [Re-enabling a Disabled Provisioning Profile](#)

### 16.3.1 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 the back-end directory.

#### Example for Oracle Internet Directory:

```
manageProvProfiles operation=create ldap_host=myhost.mycompany.com ldap_port=389
\
ldap_user_dn="cn=orcladmin"
application_dn="orclApplicationCommonName=PORTAL,cn=Portal,cn=Products,cn=OracleC
ontext" \
organization_dn="dc=us,dc=mycompany,dc=com" interface_name=PORTAL.WWSEC_OID_SYNC
\
interface_type=PLSQL \
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
,userpassword)" \
event_subscription="GROUP:dc=us,dc=mycompany,dc=com:MODIFY(uniqueMember)" \
profile_mode=OUTBOUND

```

### Example for Oracle Unified Directory

```

manageProvProfiles operation=create ldap_host=myhost.mycompany.com ldap_port=389
\
ldap_user_dn="cn=directory manager"
application_dn="orclApplicationCommonName=PORTAL,cn=Portal,cn=Products,cn=OracleC
ontext" \
organization_dn="dc=us,dc=mycompany,dc=com" interface_name=PORTAL.WWSEC_OID_SYNC
\
interface_type=PLSQL \
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
,userpassword)" \
event_subscription="GROUP:dc=us,dc=mycompany,dc=com:MODIFY(uniqueMember)" \
profile_mode=OUTBOUND

```

## 16.3.2 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:

```

manageProvProfiles operation=modify ldap_host=myhost.mycompany.com ldap_port=389
\
ldap_user_dn="cn=orcladmin"
application_dn="orclApplicationCommonName=PORTAL,cn=Portal,cn=Products,cn=OracleC
ontext" \
organization_dn="dc=us,dc=mycompany,dc=com" \
subscription="USER:dc=us,dc=mycompany,dc=com:MODIFY(orclDefaultProfileGroup,userp
assword,mail,cn,sn)"

```

## 16.3.3 Deleting a Provisioning Profile

The following example disables a provisioning profile for the Portal application.

### Example:

```

manageProvProfiles operation=delete ldap_host=myhost.mycompany.com ldap_port=389
\
ldap_user_dn="cn=orcladmin"
application_dn="orclApplicationCommonName=PORTAL,cn=Portal,cn=Products,cn=OracleC
ontext" \
organization_dn="dc=us,dc=mycompany,dc=com"

```

## 16.3.4 Disabling a Provisioning Profile

The following example disables a provisioning profile for the Portal application.

### Example:

```
manageProvProfiles operation=disable ldap_host=myhost.mycompany.com
ldap_port=389 \
ldap_user_dn="cn=orcladmin"
application_dn="orclApplicationCommonName=PORTAL,cn=Portal,cn=Products,cn=OracleC
ontext" \
organization_dn="dc=us,dc=mycompany,dc=com"
```

### 16.3.5 Re-enabling a Disabled Provisioning Profile

The following example re-enables a disabled provisioning profile for the Portal application.

**Example:**

```
manageProvProfiles operation=enable ldap_host=myhost.mycompany.com ldap_port=389
\
ldap_user_dn="cn=orcladmin"
application_dn="orclApplicationCommonName=PORTAL,cn=Portal,cn=Products,cn=OracleC
ontext" \
organization_dn="dc=us,dc=mycompany,dc=com"
```

## 16.4 Registering Applications for Provisioning

After you install an application and use the `manageProvProfiles` to create a provisioning profile for it, you must perform the steps described in this section to register the application for provisioning.

Complete 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 back-end 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 the back-end 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 the back-end 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 the back-end 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 the back-end 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 back-end 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 Unified Directory or 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`.

### Example 16-1 Oracle Internet Directory

```
dn: cn=FILES,cn=Products,cn=OracleContext
changetype: add
objectclass: orclContainer

dn: orclApplicationCommonName=Files-App1,cn=FILES,cn=Products,cn=OracleContext
changetype: add
objectclass: orclApplicationEntityorclappfullname: 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)
```



**Example 16-2 Oracle Unified Directory**

```

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
aci: (target="ldap:///orclApplicationCommonName=Files-
App1,cn=FILES,cn=Products,cn=OracleContext">ldap:///
orclApplicationCommonName=Files-App1,cn=FILES,cn=Products,cn=OracleContext"</a>)
(version 3.0; acl "Access by the DIP admin Group";
allow
(browse,proxy) groupdn="ldap:///cn=odisgroup,cn=DIPAdmins,cn=Directory">ldap:///
cn=odisgroup,cn=DIPAdmins,cn=Directory</a> Integration
Platform,cn=Products,cn=OracleContext";)
aci: (target="ldap:///orclApplicationCommonName=Files-
App1,cn=FILES,cn=Products,cn=OracleContext">ldap:///
orclApplicationCommonName=Files-App1,cn=FILES,cn=Products,cn=OracleContext"</a>)
(version 3.0; acl "Access by the User Provisioning Admins group";
allow (browse,proxy) groupdn="ldap:///cn=User">ldap:///
cn=User</a> Provisioning Admins,cn=Groups,cn=OracleContext";)
aci: (targetattr="*") (version 3.0; acl "Access to attributes
of the Application by DIP admin groups group"; allow (search,read,write,compare)
groupdn="ldap:///cn=odisgroup,cn=DIPAdmins,cn=Directory">ldap:///
cn=odisgroup,cn=DIPAdmins,cn=Directory</a> Integration
Platform,cn=Products,cn=OracleContext";)
aci: (targetattr="*") (version 3.0; acl "Access to attributes
of the Application by DIP admin groups group"; allow (search,read,write,compare)
groupdn="ldap:///cn=User">ldap:///cn=User</a> Provisioning
Admins,cn=Groups,cn=OracleContext";)

```

When you install the second instance of an application, you must create in the back-end 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`.

**Example 16-3 Oracle Internet Directory**

```

dn: orclApplicationCommonName=Files-App2,cn=FILES,cn=Products,cn=OracleContext
changetype: add
objectclass: orclApplicationEntityorclappfullname: Files Application Instance 2
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)

```

**Example 16-4 Oracle Unified Directory**

```

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 application instance.
protocolInformation: protocol information
aci: (target="ldap:///orclApplicationCommonName=Files-App2,cn=FILES,cn=Products,cn=OracleContext">ldap:///orclApplicationCommonName=Files-App2,cn=FILES,cn=Products,cn=OracleContext"</a>)
(version 3.0; acl "Access by the DIP admin Group";
allow
(browse,proxy) groupdn="ldap:///cn=odisgroup,cn=DIPAdmins,cn=Directory">ldap:///cn=odisgroup,cn=DIPAdmins,cn=Directory</a> Integration Platform,cn=Products,cn=OracleContext");
aci: (target="ldap:///orclApplicationCommonName=Files-App2,cn=FILES,cn=Products,cn=OracleContext">ldap:///orclApplicationCommonName=Files-App2,cn=FILES,cn=Products,cn=OracleContext"</a>)
(version 3.0; acl "Access by the User Provisioning Admins group";
allow (browse,proxy) groupdn="ldap:///cn=User">ldap:///cn=User</a> Provisioning Admins,cn=Groups,cn=OracleContext");
aci: (targetattr="*") (version 3.0; acl "Access to attributes of the Application by DIP admin groups group"; allow (search,read,write,compare) groupdn="ldap:///cn=odisgroup,cn=DIPAdmins,cn=Directory">ldap:///cn=odisgroup,cn=DIPAdmins,cn=Directory</a> Integration Platform,cn=Products,cn=OracleContext");
aci: (targetattr="*") (version 3.0; acl "Access to attributes of the Application by DIP admin groups group"; allow (search,read,write,compare) groupdn="ldap:///cn=User">ldap:///cn=User</a> Provisioning Admins,cn=Groups,cn=OracleContext");

```

After you successfully register a provisioned-integrated application with the back-end directory, you may need to add the application to various privileged groups. [Table 16-1](#) lists common privileged groups in Oracle Unified Directory or Oracle Internet Directory.

**Table 16-1 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

```

## 16.5 Understanding How to Configure 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 Internet Directory Provisioning supports three versions of provisioning profiles: 1.1, 2.0, and 3.0 and Oracle Unified Directory supports two versions of provisioning profiles: 2.0 and 3.0. 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 `manageProvProfiles` 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 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 Identity Management*

## 16.6 manageProvProfiles Command

You can create a provisioning integration profile using the `manageProvProfiles` command located in the `ORACLE_HOME/bin` directory.

### Syntax for manageProvProfiles

```
manageProvProfiles operation=[create|modify] ldap_host=backend_hostname
ldap_port=port
ldap_user_dn="bindDN"
[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
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

manageProvProfiles {operation=enable|disable|delete|status|reset}
application_dn=DN [organization_dn=DN] [ldap_host=backend_hostname]
[ldap_port=port]
[ldap_user_dn=bindDN] [profile_debug=debug_level]
```

### Arguments for manageProvProfiles

operation=create | modify | enable | disable | delete | status | reset

Required. The operation to perform using `manageProvProfiles`. 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=*backend\_hostname*

Optional. The host name of the Oracle Unified Directory or 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 the back-end directory. The default port for Oracle Unified Directory or Oracle Internet Directory is 389.

Required. The DN of the superuser or a user that has sufficient permissions to perform provisioning subscription operations. The default value for Oracle Unified Directory is "cn=directory manager" and for Oracle Internet Directory is "cn=orcladmin".

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 Unified Directory or 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. The default value is `TRUE`.

 **Note:**

This argument is for Oracle Delegated Administration Services 10g Releases (10.1.4.x).

manage\_application\_default=TRUE | FALSE

Optional. Determines whether the Oracle Internet Directory manages the application's default values. The default value is `TRUE`.

 **Note:**

This argument is for Oracle Delegated Administration Services 10g Releases (10.1.4.x).

```
enable_bootstrap=TRUE | FALSE
```

Optional. Indicates whether the application should receive provisioning events for users that existed in the back-end 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.

Oracle Internet Directory supports versions 1.1, 2.0, or 3.0.

Oracle Unified Directory support versions 2.0 and 3.0.

```
interface_type=PLSQL | JAVA
```

Optional. The type of interface to which events will be propagated. The default is `PLSQL`.

 **Note:**

For `JAVA` type, only interface protocol version 3.0 is supported.

```
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 the back-end directory after which all qualifying events should be provisioned to the application. Defaults to the latest current change number.

 **Note:**

To update the `lastchangenumber` argument, you must perform the following steps:

1. Stop Oracle Directory Integration Platform using Oracle Enterprise Manager Fusion Middleware Control. See [Stopping Oracle Directory Integration Platform with Fusion Middleware Control](#).
2. Update the last change number of the provisioning profile using the `manageProvProfiles` command.
3. Start Oracle Directory Integration Platform using Oracle Enterprise Manager Fusion Middleware Control. See [Starting Oracle Directory Integration Platform with Fusion Middleware Control](#).

If you do not set the `lastchangenumber` and restart Oracle Directory Integration Platform, then the `lastchangenumber` will be overwritten by an older (lower) value from the cache.

`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 the back-end 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_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.

#### Note:

For security reasons, the `ldap_user_password` and `interface_connect_info` arguments are no longer accepted on the command line.



# 17

## Understanding the Oracle Provisioning Event Engine

The Oracle provisioning event engine sends events, depending on the operation performed on the user entries in back-end directory.

### Topics:

- [What Are the Oracle Provisioning Events?](#)
- [Working with the Oracle Provisioning Event Engine](#)

### 17.1 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 back-end 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.

### 17.2 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 the back-end 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:

- [Create Custom Event Object Definitions](#)
- [Define Custom Event Generation Rules](#)

## 17.2.1 Create Custom Event Object Definitions

The Oracle provisioning event-engine provides properties that you can use to identify objects for which events can be generated.

[Table 17-1](#) lists the properties that you can use to identify objects for which events can be generated.

**Table 17-1 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 17-2](#) lists the predefined objects for which the Oracle provisioning event engine can generate events.

**Table 17-2 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

## 17.2.2 Define 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 17-3](#) lists the event definitions that are supported by the Oracle provisioning event engine.

**Table 17-3 Supported Event Definitions**

Object Name	Change Type	Rule	Event Name
USER	Add	OrclApplnUserProvStatus=PENDING_UPGRADE	USER_ADD
USER	Add	OrclApplnUserProvStatus=PROVISIONING_REQUIRED	USER_ADD
USER	Modify	OrclApplnUserProvStatus= PENDING_UPGRADE	USER_ADD
USER		OrclApplnUserProvStatus=PROVISIONING_REQUIRED	USER_ADD
USER		OrclApplnUserProvStatus=PROVISIONING_FAILURE	USER_ADD
USER		OrclApplnUserProvStatus=DEPROVISIONING_REQUIR ED	USER_MODIFY
USER		OrclApplnUserProvStatus=PROVISIONING_IN_PROGR ESS	USER_MODIFY
USER		OrclApplnUserProvStatus=PROVISIONING_SUCCESSE FUL	USER_MODIFY
USER			
USER	Delete	OrclApplnUserProvStatus=PROVISIONING_IN_PROGR ESS	USER_DELETE
USER		OrclApplnUserProvStatus=PROVISIONING_SUCCESSE FUL	USER_DELETE
USER		OrclApplnUserProvStatus=DEPROVISIONING_REQUIR ED	
GROUP	Add		GROUP_ADD
GROUP	Modify		GROUP_MODIFY
GROUP	Delete		GROUP_DELETE
IDENTITY	Add		IDENTITY_ADD

**Table 17-3 (Cont.) Supported Event Definitions**

Object Name	Change Type	Rule	Event Name
IDENTITY	Modify		IDENTITY_MODIFY
IDENTITY	Delete		IDENTITY_DELETE
ENTRY	Add		ENTRY_ADD
IDENTITY	Modify		ENTRY_MODIFY
IDENTITY	Delete		ENTRY_DELETE
SUBSCRIPTION	Add		SUBSCRIPTION_ADD
SUBSCRIPTION	Modify		SUBSCRIPTION_MODIFY
SUBSCRIPTION	Delete		SUBSCRIPTION_DELETE
SUBSCRIBER	Add		SUBSCRIBER_ADD
SUBSCRIBER	Modify		SUBSCRIBER_MODIFY
SUBSCRIBER	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.

# 18

## Integration of Provisioning Data with Oracle E-Business Suite

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 Oracle E-Business Suite documentation on the Oracle Technology Network located at:

[http://docs.oracle.com/cd/E26401\\_01/index.htm](http://docs.oracle.com/cd/E26401_01/index.htm)

# Part VI

## 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:

- [Connected Directory Integration Concepts and Considerations](#)
- [Configuring Synchronization with a Connected Directory](#)
- [Integrating with Microsoft Active Directory](#)
- [Deploying the Oracle Password Filter for Microsoft Active Directory](#)
- [Integrating with Oracle Directory Server Enterprise Edition \(Connected Directory\)](#)
- [Integrating with IBM Tivoli Directory Server](#)
- [Integrating with Novell eDirectory or OpenLDAP](#)
- [Managing Integration with a Connected Directory](#)

# 19

## Connected Directory Integration Concepts and Considerations

This chapter discusses the basic concepts of integrating Oracle Directory Integration Platform with a connected directory along with various decisions to be made as part of the integration process.

This chapter contains the following topics:

- [Concepts and Architecture of Connected Directory Integration](#)
- [Planning Your Integration Environment](#)
- [Microsoft Active Directory Integration Concepts](#)
- [Oracle Directory Server Enterprise Edition \(Sun Java System Directory Server\) Integration Concepts](#)
- [IBM Tivoli Directory Server Integration Concepts](#)
- [Novell eDirectory and OpenLDAP Integration Concepts](#)
- [Limitations of Connected Directory Integration in Oracle Directory Integration Platform](#)

### See Also:

The following chapters for specific implementation details on synchronizing with connected directories:

- [Configuring Synchronization with a Connected Directory](#)
- [Integrating with Microsoft Active Directory](#)
- [Deploying the Oracle Password Filter for Microsoft Active Directory](#)
- [Integrating with Oracle Directory Server Enterprise Edition \(Connected Directory\)](#)
- [Integrating with IBM Tivoli Directory Server](#)
- [Integrating with Novell eDirectory or OpenLDAP](#)
- [Managing Integration with a Connected Directory](#)

### 19.1 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 one of Oracle Directory, such as Oracle Unified Directory, Oracle Internet Directory, or Oracle

Directory Server Enterprise Edition 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 Directory to synchronize with a connected directory.

This section discusses the Oracle components and architecture involved in integrating Oracle Directory Integration Platform with connected directories. It included the following 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 Fusion Middleware Supported System Configurations](#) certification matrix 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 also configure a pluggable database as the connected directory.

## 19.1.1 Oracle Identity Management Components for Integrating with Other Directories

Oracle Identity Management includes Oracle Back-end Directory, Oracle Directory Integration Platform, Oracle Directory Services Manager, and Delegated Authentication that are used to integrate with another directory

**Topics:**

- [Understanding Oracle Back-end Directory](#)
- [Understanding Oracle Directory Integration Platform](#)
- [Understanding Oracle Directory Services Manager](#)
- [Understanding Delegated Authentication](#)

 **See Also:**

[Administering Oracle Directory Integration Platform](#) for a description of the tools used to integrate Oracle Internet Directory with a third-party directory

### 19.1.1.1 Understanding 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 verifying the credentials entered by users with the credentials stored in the Oracle back-end directory. Credentials can be either synchronized from the connected directory to a Oracle back-end directory or stored in the connected directory. In this case, the Oracle back-end directory will delegate the authentication to the connected directory.

### 19.1.1.2 Understanding 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 and other connected directories and user repositories.
- Automatic provisioning services for Oracle components if Oracle Internet Directory or Oracle Unified 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:

Oracle Directory Integration Platform supports password synchronization to connected directories from an Oracle back-end directory.

- 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](#)" for a discussion about configuring attribute mapping rules

### 19.1.1.3 Understanding Oracle Directory Services Manager

For more information, see [Using Oracle Directory Services Manager for Oracle Internet Directory](#).

### 19.1.1.4 Understanding Delegated Authentication

External authentication plug-ins, such as the Microsoft Active Directory external authentication plug-in, are available for the back-end directory and enable users to log in to the Oracle environment by using their Microsoft Windows credentials.

Oracle Unified Directory and Oracle Directory Server Enterprise Edition back-end directories use pass-through authentication for passing authentication through to a

connected directory like Microsoft Active Directory for users coming from 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.

For more information, see:

- The section "Understanding Pass-Through Authentication" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Unified Directory*.
- The section "[Pass-Through Authentication](#)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.
- 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.

## 19.1.2 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 in the following sections:

- [Oracle Back-end Directory Schema Elements for Microsoft Active Directory](#)
- [Understanding Oracle Internet Directory Schema Elements for Oracle Directory Server Enterprise Edition \(Sun Java System Directory Server\)](#)
- [Understanding Oracle Back-end Directory Schema Elements for IBM Tivoli Directory Server](#)
- [Understanding Oracle Back-end Directory Schema Elements for Novell eDirectory](#)
- [Understanding Oracle Back-end Directory Schema Elements for OpenLDAP](#)

## 19.1.3 Directory Information Tree in an Integration with a Connected Directory

The directory information tree (DIT) provides a way to refer to the data stored in your directory. The types of information stored, the physical nature of your enterprise, the applications that use your directory, and the types of replication you use shape the design of the directory tree.

This section contains these topics:

- [About Realms in Oracle Internet Directory](#)
- [About Suffixes in Oracle Unified Directory and Oracle Directory Server Enterprise Edition](#)
- [Understanding How to Plan the Deployment](#)
- [Example: Integration with a Single Connected Directory Domain](#)

 **See Also:**

- Understanding the Concepts and Architecture of Oracle Internet Directory in *Oracle Fusion Middleware Administering Oracle Internet Directory* for information on directory information trees.
- *Planning, Deploying and Managing Realms* section in *Oracle Fusion Middleware Administering Oracle Internet Directory* for information on the deployment of identity management realms.

### 19.1.3.1 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.

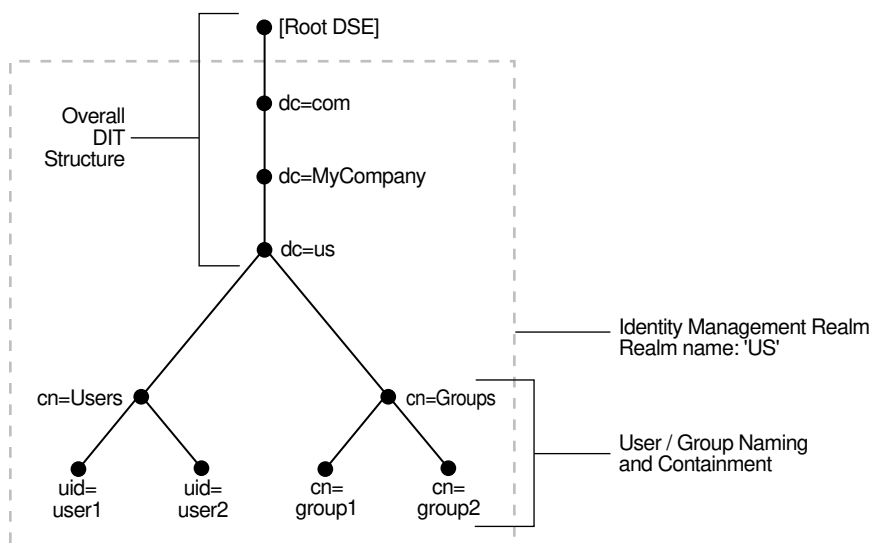
Each identity management realm is uniquely named to distinguish it from other realms. It also has a realm-specific administrator with complete administrative control over the realm.

#### The Default Realm

For all Oracle components to function, an identity management realm is required. One particular realm, created during installation of Oracle 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 19-1](#) illustrates the default identity management realm.

**Figure 19-1 The Default Identity Management Realm**

As [Figure 19-1](#) 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.

### 19.1.3.2 About Suffixes in Oracle Unified Directory and Oracle Directory Server Enterprise Edition

You must create the required suffixes in Oracle Unified Directory and Oracle Directory Server Enterprise Edition, if they do not exist.

For Oracle Directory Server Enterprise Edition you must manually create the suffixes as described in [Creating Oracle Directory Server Enterprise Edition Suffixes](#).

For Oracle Unified Directory, if the suffixes are not created during installation, then you must create the suffixes as described in [Creating Oracle Unified Directory Suffixes](#).

Ensure that you add the required Access Control Instructions (ACIs), as described in:

- [Adding Access Control Instructions \(ACIs\) for Oracle Unified Directory](#).
- [Adding Access Control Instructions \(ACIs\) for Oracle Directory Server Enterprise Edition](#).

### 19.1.3.3 Understanding How to Plan the Deployment

When planning the deployment, 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](#)".

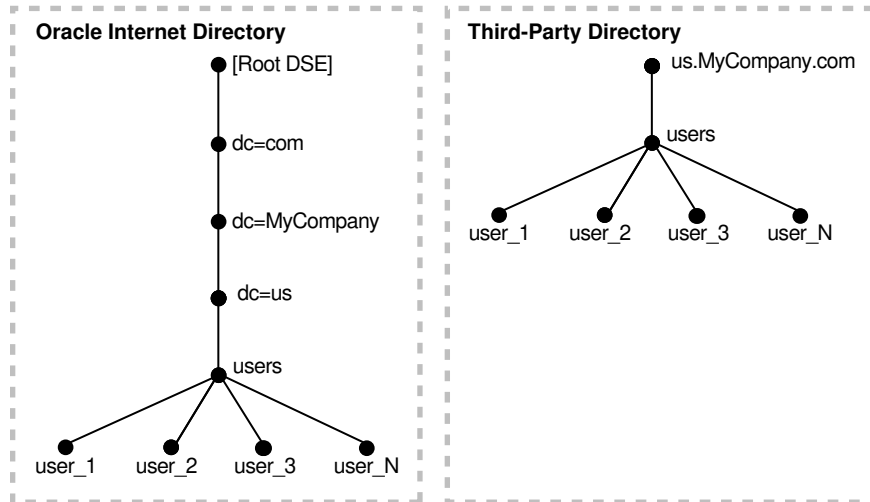
#### See Also:

The section "[About Choosing the Structure of the Directory Information Tree](#)" for more information about planning the directory information tree

### 19.1.3.4 Example: Integration with a Single Connected Directory Domain

Figure 19-2 shows an example of one-to-one mapping between Oracle Internet Directory (You can also use Oracle Unified Directory or Oracle Directory Server Enterprise Edition) and a connected directory.

**Figure 19-2 Default DIT Structures in Oracle Internet Directory and a Connected Directory When Both Directory Hosts Are Under the Domain us.MyCompany.com**



In the one-to-one mapping illustrated in Figure 19-2:

- 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 19-2, only the `users` subtree must be synchronized from the connected directory to Oracle Internet Directory using one-to-one domain mappings.

 **Note:**

In [Figure 19-2](#), 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.

## 19.2 Planning Your Integration Environment

This section describes how to plan your integration environment for Oracle Directory Integration Platform with a connected directory.

**Topics:**

- [Preliminary Considerations for Integrating with a Connected Directory](#)
- [Choose the Directory for the Central Enterprise Directory](#)
- [Understanding How to Customize the LDAP Schema](#)
- [About Choosing Where to Store Passwords](#)
- [About Choosing the Structure of the Directory Information Tree](#)
- [Attribute for the Login Name](#)
- [Selecting the User Search Base](#)
- [What Group Search Base to Select?](#)
- [Guidelines to Address Security Concerns](#)
- [About How to Administer Your Deployment with Oracle Access Manager](#)

### 19.2.1 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 or Oracle Unified 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 or Oracle Unified Directory to support Enterprise User Security. Oracle Directory Server Enterprise Edition back-end directory does 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.

## 19.2.2 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*.

**Topics:**

- [Overview of Scenario 1: Oracle Directory as the Central Enterprise Directory](#)
- [Overview of Scenario 2: A Directory Other than the Oracle Directory is the Central Enterprise Directory](#)

### 19.2.2.1 Overview of Scenario 1: Oracle Directory as the Central Enterprise Directory

If Oracle directory like Oracle Unified Directory, Oracle Internet Directory, and Oracle Directory Server Enterprise Edition is the central directory, then once the user, group, and realm objects are created, Oracle directory becomes the source of 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 synchronized to the connected directories from the Oracle directory.

[Table 19-1](#) describes the typical requirements in this deployment.

**Table 19-1 Typical Requirements with Oracle Directory as the Central Enterprise Directory**

Requirement	Description
Initial startup	The syncProfileBootstrap command populates the connected directory with users and groups stored in Oracle directory.



**Table 19-1 (Cont.) Typical Requirements with Oracle Directory as the Central Enterprise Directory**

Requirement	Description
Synchronization	User and group information is managed in Oracle 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 directory can be achieved by configuring an import profile.
Passwords synchronization	Passwords are managed in Oracle Directory by using Oracle tools. 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 as described in <a href="#">Password Synchronization</a> . Oracle recommends you to use SSL communication for password synchronization.

New users or groups in Oracle Directory can be automatically synchronized by the Oracle Directory Integration Platform. This automatic synchronization requires that the Oracle directory server is running with the change log enabled and the change log is not purged.

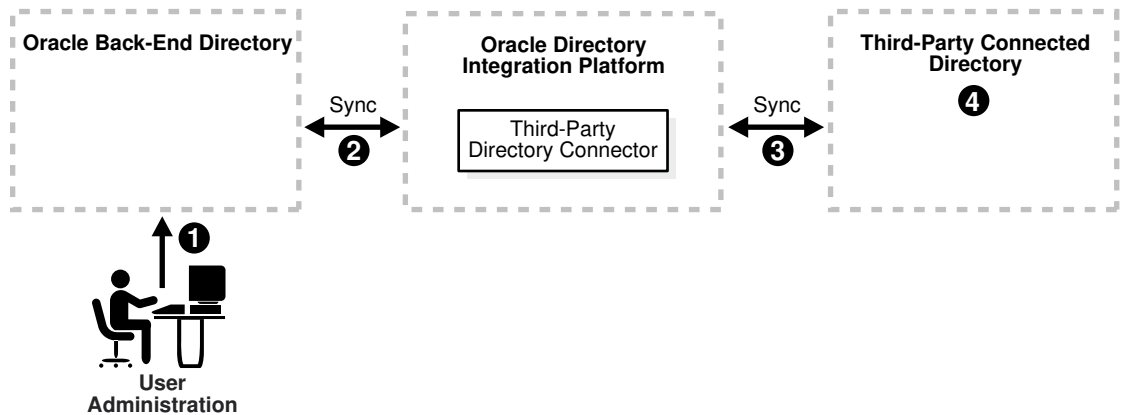
If these two conditions are not met, then you must dump the entries in Oracle Directory to an LDIF file and upload the data to the connected directory.

 **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.
- "Replicating Directory Data" in the *Oracle Fusion Middleware Administering Oracle Unified Directory*.
- "[Directory Server Replication](#)" chapter in *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.

[Figure 19-3](#) shows a typical deployment in which Oracle Directory is the central enterprise directory.

**Figure 19-3 Interaction Among Components with Oracle Directory as the Central Enterprise Directory**



As [Figure 19-3](#) shows, when Oracle Directory is the central enterprise directory, typical synchronization of a user or group follows this process:

1. The user or group entry is created in Oracle Directory by using the graphical tools or command-line tools. For more information, see [Administering Oracle Directory Integration Platform](#).
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 Directory, when a new attribute gets added to the entry, the value of an existing attribute is modified, or an existing attribute is deleted.

When Oracle 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 Directory Services Manager (For Oracle Unified Directory and Oracle Internet Directory), Directory Service Control Center (For Oracle Directory Server Enterprise Edition) 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 Directory is appropriately mapped to the corresponding attribute in the connected directory.
4. The user entry is modified in the connected directory.

## 19.2.2.2 Overview of Scenario 2: A Directory Other than the Oracle Directory is the Central Enterprise Directory

In this scenario the connected directory (Either a third-party directory or another Oracle directory) is the central enterprise directory, and Oracle 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 synchronizing information for all Oracle components and other directories. Oracle Directory is deployed as the Oracle back-end directory to support Oracle components. To provide this support, Oracle Directory stores a footprint that enables it to identify entries in the connected directory.

### Topics:

- [Understanding the Requirements in This deployment](#)
- [What are the Process for Synchronizing of a User or Group?](#)
- [What are the Process for Modifying a User or Group Entry?](#)

### 19.2.2.2.1 Understanding the Requirements in This deployment

[Table 19-2](#) describes the typical requirements in this deployment.

**Table 19-2 Typical Requirements if a Connected Directory is the Central Enterprise Directory**

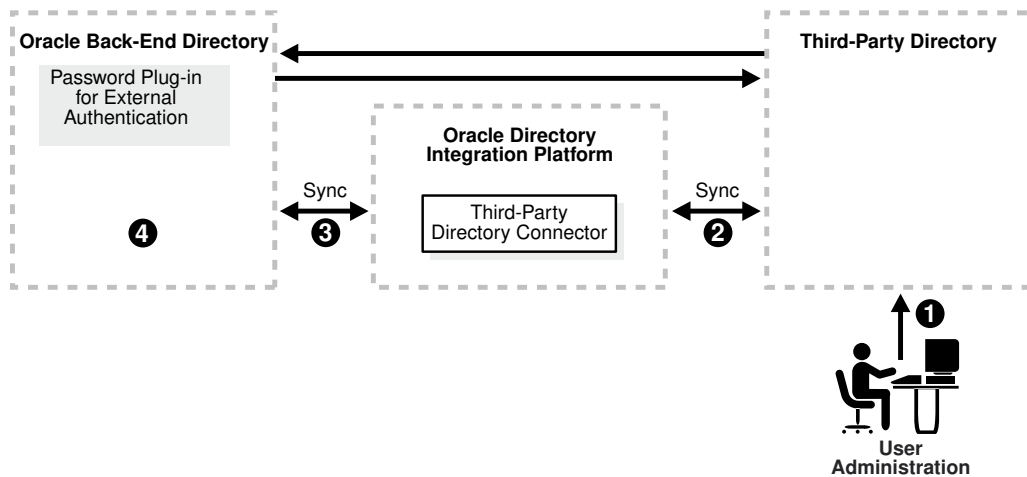
Requirement	Description
Initial startup	<p>The syncProfileBootstrap command populates Oracle back-end directory with users and groups stored in the central enterprise directory.</p> <p>You can choose to manage user information, including password credentials, in the central enterprise directory only. 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.</p> <p>Passwords are not migrated from the central enterprise directory to Oracle Internet Directory.</p>
Synchronization	<p>The central enterprise directory for user and group information is the connected directory (Oracle Internet Directory, 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 back-end directory by the Oracle Directory Integration Platform when an import profile has been configured.</p> <p>Synchronization from Oracle back-end directory to the central enterprise directory is achieved by configuring an export profile.</p>
Passwords synchronization	<p>Passwords are managed in the central enterprise directory. For more information, see <a href="#">Password Synchronization</a>.</p>
Third-party delegate authentication plug-in	<p>When user credentials are managed in the connected directory, the Oracle back-end directory can delegate the authentication. To authenticate a user, a specific plug-in performs the authentication of the user against the user credentials stored in the connected directory. For more information, see <a href="#">Understanding Delegated Authentication</a>.</p>

New users or groups created in the directory that is designated as the central enterprise directory are automatically synchronized into Oracle Directory by the Oracle Directory Integration Platform. Before this can happen, a one-way synchronization

between the central enterprise directory and Oracle back-end directory must be established.

Figure 19-4 shows a typical deployment where a third-party directory is the central enterprise directory.

**Figure 19-4 Interaction of Components with a Third-Party Directory as the Central Enterprise Directory in a Delegated Authentication Deployment**



#### 19.2.2.2.2 What are the Process for Synchronizing of a User or Group?

Figure 19-4 shows the typical process for synchronizing a user or group when a third-party directory is the central enterprise directory.

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 the Oracle back-end directory.

#### 19.2.2.2.3 What are the 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, or 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 the back-end directory.
4. The user or group entry is modified in the Oracle back-end directory.

Depending on the password synchronization method selected, when a third-party directory is the central enterprise directory, modification of passwords can happen asynchronously in the directory that serves as the password repository, as shown in [Figure 19-4](#). This happens by using plug-ins.

## 19.2.3 Understanding How to Customize the LDAP Schema

You can customize the LDAP schema in the destination directory if a directory deployment contains schema extensions such as custom object classes and attributes or 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.
- The chapter "Managing Directory Schema" in the *Oracle Fusion Middleware Administering Oracle Unified Directory*.
- The chapter "[Directory Server Schema](#)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.
- Microsoft documentation available at <http://msdn.microsoft.com/> for instructions on customizing the schema in Microsoft Active Directory.

## 19.2.4 About Choosing 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:

- [Understanding the Advantages and Disadvantages of Storing the Password in One Directory](#)
- [Understanding the Advantages and Disadvantages of Storing Passwords in Both Directories](#)

### 19.2.4.1 Understanding the 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 directory in which the passwords are stored fails, then the bind operation for users connecting to other directory also fails and the user is not authenticated.

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 configuring the delegated authentication on the back-end directory.

#### 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.

### 19.2.4.2 Understanding the Advantages and Disadvantages of Storing Passwords in Both Directories

If you decide to store passwords in both Oracle back-end directory and a connected directory, then passwords need to be synchronized, ideally in real-time.

Oracle Directory Integration Platform 11g Release 1 (11.1.1) and above does not synchronize passwords 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.

 **See Also:**

- The chapter in *Oracle Fusion Middleware Administering Oracle Internet Directory* about password policies for information about setting password policies
- The chapter in *Oracle Fusion Middleware Administering 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.

 **Note:**

Oracle recommends using the SSL connection to synchronize the password for the back-end directory and the connected directory. For more information, see [Password Synchronization](#).

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 Directory. You must configure it.

In deployments where Oracle back-end 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 for information about plug-ins:

- *Advanced Administration: Directory Plug-ins in Oracle Fusion Middleware Administering Oracle Internet Directory* about the directory plug-in framework
- *Configuring a Customized External Authentication Plug-in in Oracle Fusion Middleware Administering Oracle Internet Directory* about customizing the external authentication plug-in

## 19.2.5 About Choosing the Structure of the Directory Information Tree

At installation, each directory server might or might not create 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. Oracle Unified Directory and Oracle Directory Server Enterprise Edition do not create the default DIT. When integrating with a connected directory, you can create identical DIT structures in both directories to simplify the configuration. Alternatively, you can perform domain-level mapping.

**Topics:**

- [Create Identical DIT Structures on Both Directories](#)
- [Distinguished Name Mapping and Limitations](#)

### 19.2.5.1 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*

## 19.2.5.2 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 back-end directory and the connected directory. For example, suppose that all entries under the container `dc=mydir,dc=com` must be synchronized under `dc=myOracleDir,dc=com` in Oracle back-end 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.

### Example: User Entry Mapping

Suppose that the connected directory is Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server) and that its entries have the format `uid=name,ou=people,o=iplanet.org`. Suppose further that the back-end directory is Oracle Internet Directory whose entries 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.

### 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 DN 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](#)" for instructions about how to specify a mapping rule

## 19.2.6 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 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.

 **See Also:**

[Oracle Fusion Middleware Guide to Delegated Administration for Oracle Identity Management](#) for instructions about setting the attribute for login name

## 19.2.7 Selecting 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, do one of the following:

- Either set the user search context value to cover the entire user population.
- 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

## 19.2.8 What Group Search Base to Select?

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

## 19.2.9 Guidelines 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 back-end directory and connected directories. If you do this, then all information exchanged among the directory servers is secure.

 **Note:**

To synchronize password with Microsoft Active Directory, you must use SSL communication.

- 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.

## 19.2.10 About How to Administer 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.

## 19.3 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:

- [Understanding How to Synchronize from Microsoft Active Directory to the Oracle Back-end Directory](#)
- [Requirement for Using WebDAV Protocol](#)
- [Oracle Back-end Directory Schema Elements for Microsoft Active Directory](#)
- [About Integration with Multiple Microsoft Active Directory Domain Controllers](#)
- [Synchronizing with a Multiple-Domain Microsoft Active Directory Environment](#)
- [Understanding Foreign Security Principals](#)



**See Also:**

[Integrating with Microsoft Active Directory](#)

## 19.3.1 Understanding How to Synchronize 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 19-3](#) compares the two approaches.

**Table 19-3 Comparing the DirSync Approach to the USN-Changed Approach**

Considerations	DirSync Approach	USN-Changed Approach
Change key	Presents changes to the <code>ObjectGUID</code> , the unique identifier of the entry	Presents changes to the distinguished name. The <code>ObjectGUID</code> is used to keep track of modifications of the DN.
Enabling the Connectors	The connector is enabled by setting the property <code>Reader</code> to <code>oracle.ldap.odip.gsi.ActiveReader</code> during profile creation.	The connector is enabled by setting the property <code>Reader</code> to <code>oracle.ldap.odip.gsi.ActiveChgReader</code> during profile creation.
Template File	The synchronization profile template file is <code>adimport.properties</code> ( <code>ORACLE_HOME/ldap/odi/conf</code> ).	The synchronization profile template file is <code>activechgimp.properties</code> ( <code>ORACLE_HOME/ldap/odi/conf</code> ).
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.

**Table 19-3 (Cont.) Comparing the DirSync Approach to the USN-Changed Approach**

Considerations	DirSync Approach	USN-Changed Approach
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 <code>USNChanged</code> 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 <a href="https://support.microsoft.com/en-us">https://support.microsoft.com/en-us</a> 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: <a href="http://msdn.microsoft.com/">http://msdn.microsoft.com/</a> 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: " <a href="#">Synchronizing with a Multiple-Domain Microsoft Active Directory Environment</a> ".
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: <ul style="list-style-type: none"> <li>• Full synchronizing to a known point</li> <li>• Updating the <code>USNChanged</code> value</li> <li>• Starting synchronization with the failover directory</li> </ul> See Also: " <a href="#">Switching to a Different Microsoft Active Directory Domain Controller in the Same Domain</a> "
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: <ul style="list-style-type: none"> <li>• You are interested in import operations only</li> <li>• The Global Catalog contains all entries and attributes to be synchronized</li> <li>• Performance of the Global Catalog is acceptable</li> </ul>

 **See Also:**

["Understanding Synchronization from the Back-end Directory to a Connected Directory"](#)

## 19.3.2 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.

## 19.3.3 Oracle Back-end Directory Schema Elements for Microsoft Active Directory

[Table 19-4](#) lists the schema elements in the Oracle back-end directory for users that are imported from Microsoft Active Directory.

**Table 19-4 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.

**Table 19-4 (Cont.) Oracle Back-end Directory Schema Elements for Microsoft Active Directory**

Schema Element	Description
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.

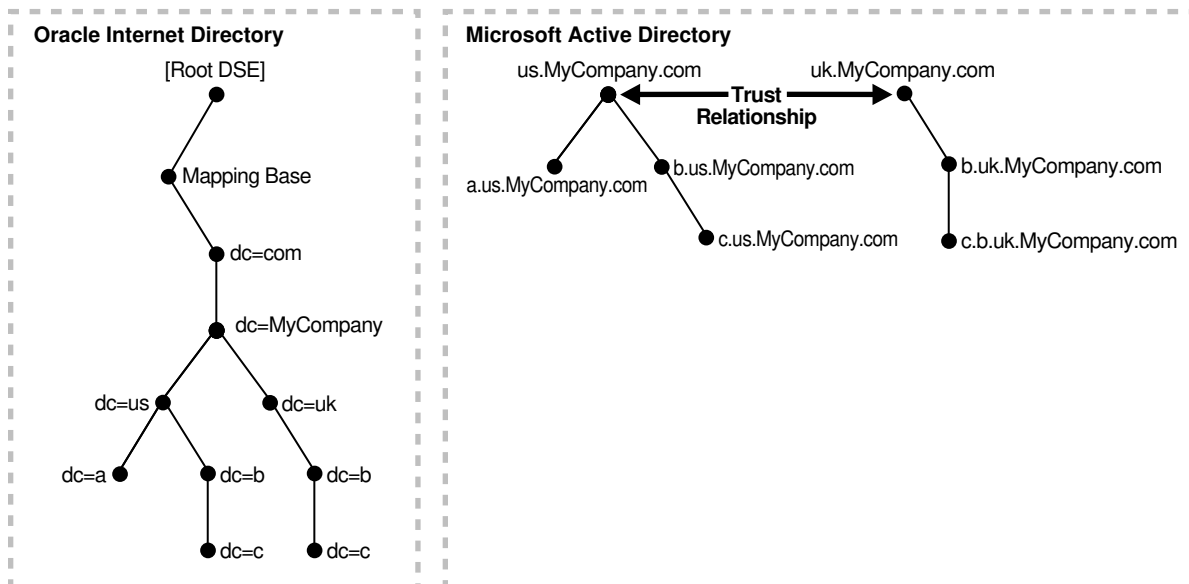
 **See Also:**

*Oracle Fusion Middleware Reference for Oracle Identity Management* for detailed information about the Oracle Internet Directory schema elements for Microsoft Active Directory.

## 19.3.4 About 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 19-5](#) shows how a forest in Microsoft Active Directory is reflected in the Oracle back-end directory.

**Figure 19-5 Mapping Between the Oracle Back-end Directory and a Forest in Microsoft Active Directory**





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 synchronizes users and groups in the respective Microsoft Active Directory domains. Before synchronization can take place, you must configure a one-way synchronization profile 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 synchronization can take place, a one-way synchronization profile between the Oracle back-end directory and a domain controller on each Microsoft Active Directory domain must be established.

## 19.3.5 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:

- [About Configuration Required for Importing from Microsoft Active Directory to the Oracle Back-end Directory](#)
- [About Configuration Required for Importing from Microsoft Active Directory Lightweight Directory Service to the Oracle Back-end Directory](#)
- [About Configuration Required for Exporting from the Oracle Back-end Directory to Microsoft Active Directory](#)
- [Example: Integration with Multiple Connected Directory Domains](#)

### 19.3.5.1 About 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

### 19.3.5.2 About 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.

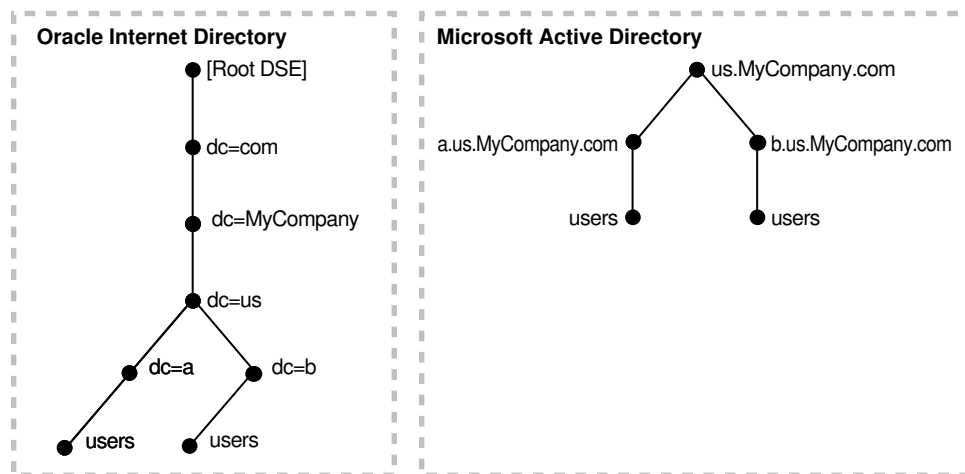
### 19.3.5.3 About Configuration Required for Exporting from the Oracle Back-end Directory to Microsoft Active Directory

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.

### 19.3.5.4 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 19-6](#) shows how multiple domains in a connected directory are mapped to a DIT in the Oracle back-end directory.

**Figure 19-6 Example of a Mapping Between the Oracle Back-end Directory and Multiple Domains in Microsoft Active Directory**



In [Figure 19-6](#), 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`.

## 19.3.6 Understanding 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.

## 19.4 Oracle Directory Server Enterprise Edition (Sun Java System Directory Server) Integration Concepts

This section contains additional considerations for integrating Oracle Internet Directory (Back-end directory) with Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server). It contains these topics:

- [Understanding Synchronization from Oracle Directory Server Enterprise Edition to Oracle Directory Integration Platform](#)
- [Understanding Oracle Internet Directory Schema Elements for Oracle Directory Server Enterprise Edition \(Sun Java System Directory Server\)](#)



### See Also:

[Integrating with Oracle Directory Server Enterprise Edition \(Connected Directory\)](#)

### 19.4.1 Understanding Synchronization 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:**

- ["Understanding Synchronization from the Back-end Directory to a Connected Directory"](#).
- "Oracle Internet Directory Administration Tools" in the *Oracle Fusion Middleware Reference for Oracle Identity Management* for instructions about how to start an Oracle 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.

## 19.4.2 Understanding 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.

## 19.5 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:

- [About Changes to Directory Objects in IBM Tivoli Directory Server](#)
- [Understanding Oracle Back-end Directory Schema Elements for IBM Tivoli Directory Server](#)

### 19.5.1 About 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.

## 19.5.2 Understanding Oracle Back-end Directory Schema Elements for IBM Tivoli Directory Server

[Table 19-5](#) lists the schema elements in the Oracle back-end directory for users that are imported from IBM Tivoli Directory Server:

**Table 19-5 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:

- ["Understanding Synchronization from the Back-end Directory to a Connected Directory"](#).
- "Oracle Internet Directory Administration Tools" in the *Oracle Fusion Middleware Reference for Oracle Identity Management* for instructions about how to start an Oracle directory server with change logging enabled.
- IBM Tivoli Directory Server documentation for instructions on how to configure change logging.

## 19.6 Novell eDirectory and OpenLDAP Integration Concepts

This section contains additional considerations for integrating the Oracle back-end directory with Novell eDirectory or OpenLDAP.

### Topics:

- [About Synchronization from Novell eDirectory or OpenLDAP to the Oracle Back-end Directory](#)
- [Understanding Oracle Back-end Directory Schema Elements for Novell eDirectory](#)
- [Understanding Oracle Back-end Directory Schema Elements for OpenLDAP](#)

### See Also:

[Integrating with Novell eDirectory or OpenLDAP](#)

## 19.6.1 About Synchronization 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:

- ["Understanding Synchronization from the Back-end Directory to a Connected Directory"](#)
- ["Customizing the Novell eDirectory or OpenLDAP Connector to Synchronize Deletions"](#) 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

## 19.6.2 Understanding Oracle Back-end Directory Schema Elements for Novell eDirectory

Know more about the schema elements in the Oracle back-end directory that are imported from Novell eDirectory.

[Table 19-6](#) lists the schema elements in the Oracle back-end directory for users that are imported from Novell eDirectory.

**Table 19-6 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.

**Table 19-6 (Cont.) Oracle Back-end Directory Schema Elements for Novell eDirectory**

Schema Element	Description
orclsourcemodifytimestamp	Required. Represents the <code>modifytimestamp</code> 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 <code>createtimestamp</code> 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:**

LDAP Schema Overview in *Oracle Fusion Middleware Reference for Oracle Identity Management* for detailed information about the Oracle back-end directory schema elements for Novell eDirectory.

## 19.6.3 Understanding Oracle Back-end Directory Schema Elements for OpenLDAP

Know more about the schema elements in the Oracle back-end directory that are imported from OpenLDAP.

[Table 19-7](#) lists the schema elements in the Oracle back-end directory for users that are imported from OpenLDAP.

**Table 19-7 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 <code>entryUUID</code> value for the respective entry in OpenLDAP. This value is used as the synchronization key.
orclsourcemodifytimestamp	Required. Represents the <code>modifytimestamp</code> attribute of the respective entry in OpenLDAP. This value is used in getting the entries that needs to be synchronized.
orclsourceCreateTimestamp	Required. Represents the <code>createtimestamp</code> attribute of the respective entry in OpenLDAP. This value is used in synchronization of deleted entries.
orclopenldapobject	Represents the OpenLDAP object.



 **See Also:**

LDAP Schema Overview in *Oracle Fusion Middleware Reference for Oracle Identity Management* for detailed information about the Oracle back-end directory schema elements for OpenLDAP.

## 19.7 Limitations of Connected Directory Integration in Oracle Directory Integration Platform

Oracle Directory Integration Platform 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:**

Schema Elements Synchronization Utility in *Oracle Fusion Middleware Reference for Oracle Identity Management* for more information about the `schemasync` tool.

# 20

## Configuring Synchronization with a Connected Directory

You can synchronize the Oracle back-end directory with a connected directory.

### 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 [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:

- [Integrating with Microsoft Active Directory](#)
- [Integrating with Oracle Directory Server Enterprise Edition \(Connected Directory\)](#)
- [Integrating with IBM Tivoli Directory Server](#)
- [Integrating with Novell eDirectory or OpenLDAP](#)

## 20.1 Verifying Synchronization Requirements

You must meet the prerequisites described in this section, before you synchronize the Oracle back-end directory with a connected directory.

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. Configure the default naming context for the connected directory.

3. 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:**

- "Oracle Internet Directory Administration Tools" in the *Oracle Fusion Middleware Reference for Oracle Identity Management* for instructions about how to start an Oracle directory server with change logging enabled.
- "orclPurgeTargetAge" in the *Oracle Fusion Middleware Reference for Oracle Identity Management* for instructions about how to set the change log purge duration.
- "Replicating Directory Data" in the *Oracle Fusion Middleware Administering Oracle Unified Directory*.
- "[Directory Server Replication](#)" chapter in *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.

4. For an OpenLDAP connected directory, ensure that Oracle Directory Integration Platform has access to read the schema definition entries (`subSchemaSubentry` attribute and `cn=Subschema` entry). Run the following command from the OpenLDAP connected directory:

- `subSchemaSubentry` attribute:

```
ldapsearch -h <OpenLDAP_Hostname> -p <OpenLDAP_Port> -D
"<SyncProfile_BindDN>" -w <PASSWORD> -b "" -s base
"(objectclass=*)" subSchemaSubentry
```

- `cn=Subschema` entry:

```
ldapsearch -h <OpenLDAP_Hostname> -p <OpenLDAP_Port> -D
"<SyncProfile_BindDN>" -w <PASSWORD> -b "cn=Subschema" -s base
"(objectclass=*)" objectClasses attributeTypes matchingRules
ldapSyntaxes objectClass javaSerializedData javaClassName
javaFactory javaCodeBase javaReferenceAddress javaClassNames
javaRemoteLocation
```

## 20.2 Creating Import and Export Synchronization Profiles Using `expressSyncSetup`

Use the `expressSyncSetup` command to create import and export synchronization profiles for express synchronization.

### Topics:

- [Understanding the `expressSyncSetup` Command](#)
- [Verifying Users and Groups Synchronization in the Connected Directory](#)

## 20.2.1 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>
```

See [expressSyncSetup command](#).

## 20.2.2 Verifying Users and Groups Synchronization in the Connected Directory

You can 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.

Run the `expressSyncSetup` command and verify that users and groups are synchronizing:

1. Run express configuration using "[Syntax for `expressSyncSetup`](#)".
2. The `expressSyncSetup` command creates two profiles named `profile_nameImport` and `profile_nameExport`. By default, both profiles are disabled. Enable the `profile_nameImport` profile if you need to synchronize from a connected directory to the Oracle back-end directory and enable the `profile_nameExport` 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 orclodioplastsuccessfulexecutiontime
```

### 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 [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 `orclodipsynchronizationstatus`.
- The value of the `Last orclodioplastsuccessfulexecutiontime` 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:

```
orclodioplastsuccessfulexecutiontime 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.

 **WARNING:**

In order to successfully customize your import and export synchronization profiles, do not enable SSL until you have finished with all other configuration tasks.

## 20.3 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](#)
- [Enable 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.

## 20.3.1 Configuring the Realm

Oracle Directory Integration Platform provides you the option to configure the realm.

### Note:

If your Oracle back-end directory is 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 "[About Choosing the Structure of the Directory Information Tree](#)", and, more specifically, in the section "[Understanding How to Plan the Deployment](#)".
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 "[Attribute for the Login Name](#)".
  - 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 19-2](#), 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`



Configure the user search base and group search base using the Oracle Directory Services Manager (For Oracle Unified Directory and Oracle Internet Directory) or Directory Service Control Center (For Oracle Directory Server Enterprise Edition).

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 19-2](#), 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.

Configure the user create base and group create base using the Oracle Directory Services Manager (For Oracle Unified Directory and Oracle Internet Directory) or Directory Service Control Center (For Oracle Directory Server Enterprise Edition).

## 20.3.2 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](#)

### 20.3.2.1 Customizing ACLs for Import Profiles

The import profile uses 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 DIP Administrator group (`cn=dipadmingrp,cn=DIPadmins,cn=Directory Integration Platform,cn=Products,cn=OracleContext`) and of the import profile group (`cn=odipigroup,cn=DIPadmins,cn=Directory Integration Platform,cn=Products,cn=OracleContext`). Those groups have appropriate privileges on `cn=oraclecontext` suffix but they need to be given appropriate privileges to perform the required actions for an import profile. 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; aci
"Entry-level DIP permissions"; allow (all,proxy)
groupdn="ldap:///cn=dipadmingrp,cn=DIPadmins,cn=Directory Integration
Platform,cn=Products,cn=OracleContext"; allow (all,proxy)
groupdn="ldap:///cn=odipigroup,cn=DIPadmins,cn=Directory Integration
```

```
Platform,cn=Products,cn=OracleContext"; )
-
add: aci
aci: (targetattr="*)(version 3.0; acl "Attribute-level DIP permissions";
allow (all,proxy) groupdn="ldap:///cn=dipadmingrp,cn=DIPadmins,cn=Directory
Integration Platform,cn=Products,cn=OracleContext"; allow (all,proxy)
groupdn="ldap:///cn=odipigroup,cn=DIPadmins,cn=Directory Integration
Platform,cn=Products,cn=OracleContext"; )
```

### 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.

You can upload an LDIF file using the following `ldapmodify` command:

```
$ORACLE_HOME/bin/ldapmodify -h <backend-host> -p <backend-port> -D <binddn> -f
<ldif-file>
```

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 11g 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*

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 <backend-host> -p <backend-port> -D <binddn> -f
<ldif-file>
```

After executing the command, you will be prompted for the password for privileged directory user.

## 20.3.2.2 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 <backend-host> -p <backend-port> -D <binddn>
-f <ldif-file>
```

## 20.3.3 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](#)" for a full discussion of mapping rules
- The section "[Supported Attribute Mapping Rules and Examples](#)" 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](#)".
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 Fusion Middleware Reference for Oracle Identity Management*.

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.

## 20.3.4 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 [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:**

- If the Oracle back-end directory is Oracle Internet Directory, then Oracle Directory Integration Platform supports both No Authentication SSL mode (SSL mode 1) and SSL Server Authentication (SSL mode 2). If Oracle Unified Directory or Oracle Directory Server Enterprise Edition is your Oracle back-end directory, then SSL Server Authentication (SSL mode 2) is the only SSL option.
  - Oracle does not recommend using No Authentication (SSL Mode 1).
- 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_JKS \
-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 3.a and 3.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_JKS
```

- 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 20-1 Supported Values for `sslmode` in `connectedDirectoryURL` Attribute**

Supported <code>sslmode</code> Value	Description
0	No SSL mode. Supported for all directory types.
1	No Authentication mode. No certificate. Supported only for Oracle Internet Directory.
<div style="border: 1px solid #0070c0; padding: 10px; background-color: #e6f2ff;">  <b>Note:</b> Oracle does not recommend using No Authentication (SSL Mode 1). </div>	
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 [Managing the SSL Certificates of Back-End Directories and Connected Directories](#).
- Oracle Directory Integration Platform 12c supports Transport Layer Security (TLS) v1.2 protocol for communication with connected directories.  
See [Transport Layer Security Protocol and Cipher Suites](#).

## 20.3.5 Enable Password Synchronization from the Oracle Back-end Directory to a Connected Directory

For more information, see [Password Synchronization](#).

## 20.3.6 Configuring External Authentication Plug-ins

Oracle Internet Directory 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:

 **Note:**

Do not change the default attribute names for the external authentication plug-in. For example, changing the default attribute name for `orclpluginflexfield` will throw an error.



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

## 20.3.7 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 20-2](#). You can use `*` as a wildcard, as shown in the example.

 **Note:**

For multiple Microsoft Active Directory domains or server external authentication plug-ins, you must synchronize each Microsoft Active Directory server set of users into separate Oracle Internet Directory containers or trees and ensure that appropriate DN is listed in the corresponding plug-in's `orclpluginsubscriberdnlist` value.

For example, default plug-ins

like `cn=oidexplg_bind_ad,cn=plugin,cn=subconfigsubentry`  
and `cn=oidexplg_compare_ad,cn=plugin,cn=subconfigsubentry`  
must include

`orclpluginsubscriberdnlist=cn=ad1,cn=users,dc=mycompany,dc=com.`

New plug-ins like

`cn=oidexplg_bind_ad2,cn=plugin,cn=subconfigsubentry` and  
`cn=oidexplg_compare_ad2,cn=plugin,cn=subconfigsubentry`  
must have

`orclpluginsubscriberdnlist=cn=ad2,cn=users,dc=mycompany,dc=com.`

You can keep the default names for the default plug-ins and provide a unique name for the new plug-ins.

**Table 20-2 Distinguished Names of External Authentication Plug-ins**

Plug-in Type	DN
Microsoft Active Directory	<code>cn=oidexplg_compare_ad, cn=plugin,cn=subconfigsubentry cn=oidexplg_bind_ad, cn=plugin,cn=subconfigsubentry</code>
Oracle Directory Server Enterprise Edition (Sun Java System Directory Server)	<code>cn=oidexplg_compare_iplanet, cn=plugin,cn=subconfigsubentry cn=oidexplg_bind_iplanet, cn=plugin,cn=subconfigsubentry</code>
Novell eDirectory	<code>cn=oidexplg_compare_Novell eDirectory, cn=plugin,cn=subconfigsubentry cn=oidexplg_bind_Novell eDirectory, cn=plugin,cn=subconfigsubentry</code>
OpenLDAP	<code>cn=oidexplg_compare_openldap, cn=plugin,cn=subconfigsubentry cn=oidexplg_bind_openldap, cn=plugin,cn=subconfigsubentry</code>

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: myhost.us.example.com
orclpluginflexfield/port2: 636
orclpluginflexfield/issl2: 1
orclpluginflexfield/host: myhost.us.example.com
orclpluginflexfield/walletloc2: /location/wallet
orclpluginflexfield/port: 389
orclpluginflexfield/walletloc: /tmp
orclpluginflexfield/issl: 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: orclPluginConfigobjectclass: 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/issl: 0
orclpluginflexfield/isfailover: 0
orclpluginflexfield/host2: myhost.us.example.com
orclpluginflexfield/port2: 636
orclpluginflexfield/issl2: 1
orclpluginflexfield/host: myhost.us.example.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 output, 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.

## 20.4 Writing Custom Synchronization Connectors

Oracle Directory Integration Platform supports custom synchronization connectors. This section provides information to help you write custom connectors.

### Topics:

- [Writing Inbound Connectors](#)
- [Writing an Outbound Connectors](#)

### 20.4.1 Writing Inbound Connectors

You can create write an inbound connector for custom synchronization 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 for Inbound Connectors](#)" 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](#)" 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.

## 20.4.2 Writing an Outbound Connectors

You can create write an outbound connector for custom synchronization 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 for Outbound Connectors](#)" 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](#)" 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.

## 20.5 Sample Reader for Inbound Connectors

Following example shows a sample reader for inbound connectors.

```
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.Utills;
//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.

    }
}
}
```

## 20.6 Sample Writer for Outbound Connectors

Following example shows a sample writer for Outbound Connectors.

```
*/

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.Utills;

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_MODALDN:  
        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 ODIEException {
    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 ODIEException {
    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
}

```

## 20.7 expressSyncSetup command

The `expressSyncSetup` command located in the `ORACLE_HOME/bin` directory allows you create import and export synchronization profiles for express synchronization.

### Note:

- 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. See *Configuring SSL in Oracle Fusion Middleware Securing Oracle WebLogic Server* for more information.
- Oracle Directory Integration Platform 12c supports Transport Layer Security (TLS) v1.2 protocol for communication with connected directories. See [Transport Layer Security Protocol and Cipher Suites](#).

### Syntax for 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]
```

## Arguments for expressSyncSetup Command

**Table 20-3** dipStatus utility Arguments

Argument	Description
<b>-h   -host</b>	Oracle WebLogic Server where Oracle Directory Integration Platform is deployed.
<b>-p   -port</b>	Listening port of the Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed.
<b>-D   -wlsuser</b>	Oracle WebLogic Server login ID. <b>Note:</b> 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 <code>expressSyncSetup</code> 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 <code>expressSyncSetup</code> , put each on a separate line in the file, in the following order: connected directory bind DN password, then Oracle WebLogic Server login password.
<b>-pf   -profile</b>	Profile name. Specify the name of the profile in ASCII characters only, as non-ASCII characters are not supported in the profile name.
<b>-conDirType</b>	Connected directory type. The supported values are ActiveDirectory, EDirectory, iPlanet, OpenLDAP, ADAM, Tivoli, OID, and ExchangeServer2003.
<b>-conDirUrl</b>	URL where the connected directory is running. The format is <i>host.port</i> .
<b>-conDirBindDN</b>	Connected directory server bind DN. For example: <ul style="list-style-type: none"> <li>• administrator@idm2003.net</li> <li>• cn=orcladmin</li> <li>• cn=Directory Manager</li> </ul> <b>Note:</b> 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 <code>expressSyncSetup</code> 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 <code>expressSyncSetup</code> , put each on a separate line in the file, in the following order: connected directory bind DN password, then Oracle WebLogic Server login password.
<b>-conDirContainer</b>	The synchronization container. For example: <ul style="list-style-type: none"> <li>• ou=sales,dc=us,dc=com</li> <li>• ou=groups,dc=imtest,dc=com</li> <li>• cn=users,dc=imtest,dc=com</li> </ul>

Table 20-3 (Cont.) dipStatus utility Arguments

Argument	Description
<b>-ssl</b>	Executes the command in SSL mode. <b>Note:</b> The Oracle WebLogic Managed Server where Oracle Directory Integration Platform is deployed must be configured for SSL to execute this command in SSL mode. For more information, see "Configuring SSL" in <i>Oracle Fusion Middleware Securing Oracle WebLogic Server</i> .
<b>-enableProfiles</b>	Specify true to enable created profiles, false if not.
<b>-keystorePath</b>	The full path to the keystore.
<b>-keystoreType</b>	The type of the keystore identified by -keystorePath. For example: -keystorePath jks or -keystorePath PKCS12
<b>-help</b>	Provides usage help for the command.
<b>-search-size-limit</b>	Specify unlimited as the entries size limit, if Oracle Directory Server Enterprise Edition is the back-end directory.

### 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
```

# 21

## Integrating with Microsoft Active Directory

This chapter outlines the procedures for integrating Oracle Identity Management with Microsoft Active Directory in a production environment.

### Topics:

- [Verify Synchronization Requirements for Microsoft Active Directory](#)
- [Configuring Basic Synchronization with Microsoft Active Directory](#)
- [Configuring Advanced Integration with Microsoft Active Directory](#)
- [Using DirSync Change Tracking for Import Operations](#)
- [Configuring Synchronization of Microsoft Active Directory Foreign Security Principal References with an Oracle Back-End Directory](#)
- [Switching to a Different Microsoft Active Directory Domain Controller in the Same Domain](#)
- [About Configuration for Microsoft Active Directory Connector with Microsoft Active Directory Lightweight Directory Service](#)
- [Configuring the Microsoft Active Directory Connector for Microsoft Exchange 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:

- [Introduction to Oracle Directory Integration Platform](#)
- [Managing the Oracle Directory Integration Platform](#)
- [Understanding the Oracle Directory Synchronization Service](#)
- [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, available on Oracle Technology Network at <http://www.oracle.com/technology/>

### 21.1 Verify 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.

You must follow the instructions in "[Verifying Synchronization Requirements](#)".

## 21.2 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 [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.

## 21.3 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

 **Note:**

- 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:**

["Customizing the Search Filter to Retrieve Information from Microsoft Active Directory"](#) 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:

- [Understanding How to Plan Integration with Microsoft Active Directory](#)
- [Configure the Realm for Microsoft Active Directory](#)
- [Customizing the Search Filter to Retrieve Information from Microsoft Active Directory](#)
- [Understanding How to Customize the ACLs for Microsoft Active Directory](#)
- [Customize Attribute Mappings for Integrating with Microsoft Active Directory](#)
- [Synchronizing with Multiple Microsoft Active Directory Domains](#)
- [About How to Synchronize Deletions from Microsoft Active Directory](#)
- [About Synchronization in SSL Mode](#)
- [Synchronizing Passwords from the Oracle back-End Directory to Microsoft Active Directory](#)
- [About the Microsoft Active Directory External Authentication Plug-in Configuration](#)
- [Perform Post-Configuration and Administrative Tasks](#)

## 21.3.1 Understanding How to Plan Integration with Microsoft Active Directory

Plan your integration by reading [Connected Directory Integration Concepts and Considerations](#), particularly "[Microsoft Active Directory Integration Concepts](#)". Be sure to create a new profile by copying the existing Active Directory template profile by following the instructions in "[Creating Synchronization Profiles](#)".

## 21.3.2 Configure the Realm for Microsoft Active Directory

If your Oracle back-end directory is Oracle Internet Directory, configure the realm by following the instructions in "[Configuring the Realm](#)".

## 21.3.3 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)(!(objectclass=computer))))
```

You can use Oracle Enterprise Manager Fusion Middleware Control to customize the search filter by completing 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/lem`.
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](#)".

#### 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

## 21.3.4 Understanding How to Customize the ACLs for Microsoft Active Directory

Customize ACLs as described in "[Customizing Access Control Lists](#)".

## 21.3.5 Customize Attribute Mappings for Integrating with Microsoft Active Directory

When integrating with Microsoft Active Directory, the following attribute-level mapping is mandatory for all objects:

```
ObjectGUID: : : :orclObjectGUID:
ObjectSID: : : :orclObjectSID:
```



**Example 21-1 Attribute-Level Mapping for the User Object in Microsoft Active Directory**

```
SAMAccountName:1 :user:orclADSAMAccountName: :orclADUser
userPrincipalName: :user:orclADUserPrincipalName::orclADUser:userPrincipalName
```

**Example 21-2 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](#)".

## 21.3.6 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 "[About Configuration Required for Importing from Microsoft Active Directory to the Oracle Back-end Directory](#)".

**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 "[Configure Connection Details for a Third-Party Directory](#)".
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.

## 21.3.7 About How to Synchronize 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 `dscls.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

## 21.3.8 About Synchronization 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](#)".

## 21.3.9 Synchronizing Passwords from the Oracle back-End Directory to Microsoft Active Directory

To synchronize password changes from the Oracle back-end directory to Microsoft Active Directory, follow these steps:

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 "[Enable Password Synchronization from the Oracle Back-end Directory to a Connected Directory](#)".

## 21.3.10 About the Microsoft Active Directory External Authentication Plug-in Configuration

Configure the Microsoft Active Directory external authentication plug-in by following the instructions in "[Configuring External Authentication Plug-ins](#)".

## 21.3.11 Perform Post-Configuration and Administrative Tasks

Read [Managing Integration with a Connected Directory](#) for information on post-configuration and ongoing administration tasks.

# 21.4 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
```

## 21.5 Configuring Synchronization of Microsoft Active Directory Foreign Security Principal References with an Oracle Back-End Directory

This section explains how to synchronize Microsoft Active Directory foreign security principal references with an Oracle back-end directory.

Although Microsoft Active Directory stores information for group members in a trusted domain as foreign security principal references, Oracle back-end directory stores the DN's of these members. 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 back-end directory.

To establish the relationship between users and groups, the member DN's that refer to the foreign security principals must be replaced by the DN's 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 and above.

## 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 21-3](#), 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: Update the Mapping Rules to Resolve the Foreign Security Principals During Synchronization

After bootstrapping, modifications to groups must be reflected in back-end 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 21-3](#), 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 21-3](#), the `domainrules` in the mapping file for Domain A should have content similar to the following:

```
DOMAINRULES
<Src Domain A >:<Dst domain A1 in back-end directory>
<Src Domain B >:<Dst domain B1 in back-end directory>
<Src Domain C >:<Dst domain C1 in back-end directory>
```

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
```

### Example 21-3 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.

## 21.6 Switching to a Different Microsoft Active Directory Domain Controller in the Same Domain

You can use the USN-Changed approach or DirSync approach to change the Microsoft Active Directory domain controller to which changes are exported.

### Topics:

- [Changing the Microsoft Active Directory Domain Controller by Using the USN-Changed Approach](#)
- [Changing the Microsoft Active Directory Domain Controller by Using the DirSync Approach](#)

### 21.6.1 Changing 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))(!
(objectclass=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 3.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.

## 21.6.2 Changing 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.

## 21.7 About Configuration for Microsoft Active Directory Connector with 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.

## 21.8 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 the Fusion Middleware Control or `manageSyncProfiles` command to configure the Microsoft Active Directory connector for Microsoft Exchange Server.

### Topics:

- [Enabling Microsoft Exchange User Synchronization Using the Fusion Middleware Control](#)
- [Enabling Microsoft Exchange User Synchronization From the Command Line](#)

To further customize your integration with Microsoft Exchange, follow the instructions in "[Configuring Advanced Integration with Microsoft Active Directory](#)".



## 21.8.1 Enabling Microsoft Exchange User Synchronization Using the Fusion Middleware Control

1. Use the Oracle Enterprise Manager Fusion Middleware Control to create a synchronization profile, as described in “[Creating Synchronization Profiles](#)”.  
 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=DADV MN0152,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.

## 21.8.2 Enabling Microsoft Exchange User Synchronization From the Command Line

1. Use the `manageSyncProfiles` command. For more information, see [manageSyncProfiles Utility](#).

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](#).

- 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 (DADV MN0152),CN=First Storage
Group, CN=InformationStore,CN=DADV MN0152,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

# 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.

## Topics:

- [Overview of the Oracle Password Filter for Microsoft Active Directory](#)
- [Understanding How to Configure and Test Oracle Back-end Directory with SSL Server-Side Authentication](#)
- [Importing a Trusted Certificate into a Microsoft Active Directory Domain Controller](#)
- [Testing SSL/TLS Communication Between Oracle Back-end directory and Microsoft Active Directory](#)
- [Installing and Reconfiguring the Oracle Password Filter for Microsoft Active Directory](#)
- [Removing the Oracle Password Filter for Microsoft Active Directory](#)

For help troubleshooting an issue with the Oracle Password Filter for Microsoft Active Directory, see the following topic in [Troubleshooting the Oracle Directory Integration Platform](#).

- [Oracle Password Filter for Microsoft Active Directory Errors and Problems](#)

### Note:

The installation file for the Oracle Password Filter for Microsoft Active Directory is located in `ORACLE_HOME/dip/utills/adpwdfilter` (UNIX) or `ORACLE_HOME\dip\utills\adpwdfilter` (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. For more information, see "[Installing the Oracle Password Filter for Microsoft Active Directory](#)".

## 22.1 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?](#)
- [Learn How the Oracle Password Filter for Microsoft Active Directory Work?](#)

- [Deploying the Oracle Password Filter for Microsoft Active Directory?](#)

## 22.1.1 What is the Oracle Password Filter for Microsoft Active Directory?

Oracle Directory Integration Platform enables synchronization between the Oracle back-end directory and Microsoft Active Directory. The Oracle Directory Integration Platform can retrieve all Microsoft Active Directory attributes with the exception of user passwords. Applications 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. 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 support 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 and Oracle Unified Directory back-end directories. 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 and Oracle Unified Directory back-end directories.

The Oracle Directory Server Enterprise Edition back-end directory do not support integration with Enterprise User Security.

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 `orclObjectGUID` 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.
- Ensure that the Microsoft Active Directory is enabled to use secure protocol TLS v1.2 or TLS v1.1. Oracle Internet Directory and Oracle Unified Directory 12c supports TLS v1.2 and TLS v1.1 protocol for communication.

You can also configure TLS v1 or SSLv3 with Oracle Internet Directory and Oracle Unified Directory 12c. Oracle does not recommend this.

## 22.1.2 Learn How 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:

- [Understanding How Clear Text Password Changes are Captured](#)
- [Understanding How Password Changes are Stored when the Oracle Back-end Directory is Unavailable](#)
- [About Delay in Password Synchronization Until Microsoft Active Directory Users are Synchronized with Oracle Back-end Directory](#)
- [Understanding Password Bootstrapping](#)

### 22.1.2.1 Understanding 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.

## 22.1.2.2 Understanding How 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.

### Note:

The Password Filter encryption is proprietary of Microsoft. Oracle Directory Integration Platform uses the `CryptProtectData` function for data encryption and provides `CRYPTPROTECT_UI_FORBIDDEN` as the flag value. The `CryptProtectData` function is associated with an user, and only the associated user can decrypt the password. For Oracle Password Filter for Microsoft Active Directory, a system user has the same identity as the LSA.

## 22.1.2.3 About Delay in Password Synchronization Until Microsoft Active Directory Users are Synchronized with Oracle Back-end Directory

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.

## 22.1.2.4 Understanding 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.

## 22.1.3 Deploying 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 and Microsoft Active Directory by following the instructions described in [Integrating with Microsoft Active Directory](#).

2. Configure and test the Oracle back-end directory in SSL server authentication mode by following the instructions in "[Understanding How to Configure and Test Oracle Back-end Directory with SSL Server-Side Authentication](#)".
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](#)".
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/TLS Communication Between Oracle Back-end directory and Microsoft Active Directory](#)".
5. Install the Oracle Password Filter for Microsoft Active Directory by following the instructions in "[Installing the Oracle Password Filter for Microsoft Active Directory](#)".
6. Configure the Oracle Password Filter for Microsoft Active Directory by following the instructions in "[Reconfiguring the Oracle Password Filter for Microsoft Active Directory](#)".

## 22.2 Understanding How to Configure and Test Oracle Back-end Directory with SSL Server-Side Authentication

Use SSL server authentication mode to synchronize password changes between back-end directory and Microsoft Active Directory.

The Oracle Password Filter communicates password changes from Microsoft Active Directory to back-end 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 back-end 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 back-end directory server certificate must match the back-end directory server hostname. For example, if the Oracle Internet Directory server hostname is `oid.example.com`, then the subject attribute of the Oracle Internet Directory server certificate must also be `oid.example.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 back-end directory and Microsoft Active Directory integration, back-end 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 back-end 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.

## 22.3 Importing a Trusted Certificate into a Microsoft Active Directory Domain Controller

You must use the Microsoft Management Console to import the certificate authority's trusted certificate into the domain controller.

Server-authenticated SSL communication between a Microsoft Active Directory domain controller and back-end directory will fail if the domain controller does not recognize the back-end directory SSL certificate as valid. In order for a domain controller to accept an back-end 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>.

## 22.4 Testing SSL/TLS Communication Between Oracle Back-end 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 or TLS communication between back-end 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 back-end 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 back-end 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 back-end 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](#)".

3. Close the command prompt window.

## 22.5 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.

### Topics:

- [Prerequisites to Install or Reconfigure the Oracle Password Filter for Microsoft Active Directory](#)
- [Installing the Oracle Password Filter for Microsoft Active Directory](#)
- [Reconfiguring the Oracle Password Filter for Microsoft Active Directory](#)

### 22.5.1 Prerequisites to Install or Reconfigure 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 back-end directories. [Table 22-1](#) lists the configuration parameters you will need for Microsoft Active Directory and [Table 22-2](#) lists the configuration parameters you will need for back-end directories.

**Table 22-1 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 <i>mycompany.com</i> .
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 <code>organizationalUnit</code> within the specified container. For this reason, the specified container should be capable of holding <code>organizationalUnit</code> objects. This value is typically in the form <i>dc=mycompany,dc=com</i> .
Port	The Microsoft Active Directory LDAP port (usually 389).
Host	The IP address (NOT the host name) of the Microsoft Active Directory domain controller.

**Table 22-1 (Cont.) Oracle Password Filter Configuration Parameters for Microsoft Active Directory**

Parameter	Description
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 <i>administrator@ad_domain.name</i> .
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 <i>E:\ADPasswordFilter\Log</i> .

**Table 22-2 Oracle Password Filter Configuration Parameters for Oracle Back-end Directory**

Parameter	Description
Base DN	The container in the back-end directory DIT where the Oracle Password Filter searches for entries synchronized from Microsoft Active Directory. For example: <i>o=Microsoft Active Directory,c=us</i> .
Host	Specifies the host name where the back-end directory LDAP processes are running. For Oracle Unified Directory and Oracle Internet Directory installations running in a high availability configuration, use the virtual host name of the load balancer. For more information, see the section "Oracle Directory Integration Platform High Availability" in the <i>Oracle Fusion Middleware High Availability Guide</i> .
SSL Port	The back-end directory port that is configured for SSL server authentication.
Non-SSL Port	The back-end directory for unencrypted communication.
User	The distinguished name of a back-end directory user with permissions to update user passwords in the base DN. For example: <i>cn=orcladmin</i> (Oracle Internet Directory) or <i>cn=Directory Manager</i> (Oracle Unified Directory or Oracle Directory Server Enterprise Edition).
User Password	The specified back-end directory user's password.

## 22.5.2 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 back-end directory configuration parameters listed in the following procedure are described in [Table 22-1](#) and [Table 22-2](#).

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 `ORACLE_HOME\dip\utils\adpwdfilter` directory in the distribution package.
- b. 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. Locate the `setup.exe` file in the `ORACLE_HOME\dip\utils\adpwdfilter\64bit` directory in the distribution package.
- b. 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.

 **Note:**

`setup.exe` is a Windows 32-bit binary built on a Windows 64-bit binary.

2. On the Welcome page, click **Next**. The Installation Requirements page is displayed, notifying you that SSL must be enabled between back-end 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 screen, click **Next**. The Installation Options screen is displayed.
4. Select **Typical (Recommended)** or **Advanced**. If you select the **Advanced** option then you can specify attributes for back-end directory and Microsoft Active Directory later in the installation process (Step 10). Click **Next**.

The Installation Location screen is displayed, prompting you for the folder where you want to install Oracle Password Filter for Microsoft Active Directory.

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 Active Directory Configuration Parameters screen is displayed.

5. Enter values for the following parameters:
  - **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 389).
  - **Host:** The IP address (NOT the host name) of the Microsoft Active Directory domain controller.

Click **Next**.

The Microsoft Active Directory Domain Controller Information screen is displayed.

6. Enter the values for the following parameters:
  - **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@ad_domain.name`.
  - **User Password:** Specify the Microsoft Active Directory user's password.
  - **Log File Path:** Accept the default location where the log files will be written or select **Browse** to locate a different directory.

7. Click **Next** to continue.

The Oracle Backend Directory Configuration Parameters page is displayed.

8. Enter values for the following parameters:
  - **Base DN:** The container in the back-end directory DIT where the Oracle Password Filter searches for entries synchronized from Microsoft Active Directory. For example: `o=Microsoft Active Directory,c=us`.

- **Host:** Specify the host name where the back-end directory LDAP processes are running. For back-end directory installations running in a high availability configuration, use the virtual host name of the load balancer.
- **SSL Port:** Enter the SSL port number for the back-end directory.
- **Non-SSL Port:** Enter the The back-end directory port number for unencrypted communication.
- **User:** The distinguished name of a back-end directory user with permissions to update user passwords in the base DN. For example: `cn=orcladmin` (Oracle Internet Directory) or `cn=Directory Manager` (Oracle Unified Directory or Oracle Directory Server Enterprise Edition).
- **User Password:** The back-end directory password.

 **Note:**

If you have configured both import and export synchronization between back-end 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 back-end directory. This is necessary to prevent password updates from looping between back-end directory and Microsoft Active Directory.

Click **Next**.

The Configuration Parameter Information screen is displayed.

9. Enter values for the following parameters:

- **SleepTime:** The number of minutes between attempts to synchronize passwords changes between back-end directory and Microsoft Active Directory.
- **ConfigSleepTime:** The number of minutes between attempts to synchronize configuration changes between back-end directory and Microsoft Active Directory.
- **ExcludeListDN:** A fully qualified DN containing a list of users whose passwords should not be synchronized.

The DLL can ignore certain entries from the password synchronization. To do so, you must add the users in the remote LDAP server under a given subtree.

ExcludeListDN needs to be configured with the same value on all Microsoft Active Directory servers where the Oracle Password Filter is installed.

Once the DLL starts, the `cn=ExcludeList` attribute will be created under the entry configured in ExcludeListDN.

You need to update entry as follows:

```
dn: cn=user2@fr.example.com,cn=ExcludeList,<ExcludeListDN>
cn: user2@fr.example.com
objectClass: orclcontainer
objectClass: top
```

In the above example:

- `user2` is the value of `samAccountName`.
- `fr.example.com` is the `ADDomain` attribute in the Windows registry.

Once the above entry is added, the password for `user2` will not be synchronized.

- **Maximum Retries:** Specifies the maximum number of attempts to synchronize a password.
10. Click **Next** to continue. If you chose Advanced on the Installation Options page, the Specify Attributes page is displayed.

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 back-end 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 is displayed and lists the path where the Oracle Password Filter for Microsoft Active Directory will be installed.
11. Click **Next** to install the Oracle Password Filter.
  12. When prompted whether or not to upload schema extensions to Oracle Backend Directory, select **Yes** if the back-end directory is Oracle Internet Directory.

For Oracle Unified Directory and Oracle Directory Server Enterprise Edition, select **No**.

The Restart page is displayed.

13. Click **Next** to restart the computer.
14. 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. Locate the following two DLL files in `C:\WINDOWS\syswow64` and copy them to `C:\WINDOWS\system32`:
  - `oraidmpwf10.dll`
  - `orclmessages.dll`
- c. Restart the Active Directory server.

The Oracle Password Filter for Microsoft Active Directory is now installed.

## 22.5.3 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 back-end directory configuration parameters listed in the following procedure are described in [Table 22-1](#) and [Table 22-2](#).

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 is displayed.

2. Click **Next**.

The Active Directory Configuration Parameters page is displayed.

3. Modify the following parameters:

- **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 389).
- **Host:** The IP address (NOT the host name) of the Microsoft Active Directory domain controller.

Click **Next**.

The Oracle Backend Directory Configuration Parameters screen is displayed.

4. Modify the following parameters:

- **Base DN:** The container in the back-end directory DIT where the Oracle Password Filter searches for entries synchronized from Microsoft Active Directory. For example: `o=Microsoft Active Directory,c=us`.
- **Host:** Specify the host name where the back-end directory LDAP processes are running. For back-end directory installations running in a high availability configuration, use the virtual host name of the load balancer.
- **SSL Port:** SSL port number of the back-end directory.

 **Note:**

At the point of reconfiguring, two configuration set entries exist in back-end directory and two instances of the back-end directory are running, each instance with one configuration set entry. Enter the SSL port of the second configuration set entry in the **SSL Port** field.

Click **Next** to continue.

The Oracle Password Filter Configuration Parameters screen is displayed.

5. Modify the following parameters:
  - **SleepTime**: The number of minutes between attempts to synchronize passwords changes between back-end directory and Microsoft Active Directory.
  - **ConfigSleepTime**: The number of minutes between attempts to synchronize configuration changes between back-end 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.

Click **Next**.

The Reconfiguration Completed Successfully page displays.

6. On the Reconfiguration Completed Successfully page, click **Finish** to reconfigure the Oracle Password Filter.

After reconfiguring the Oracle Password Filter, you must restart the Microsoft Active Directory

## 22.6 Removing the Oracle Password Filter for Microsoft Active Directory

You can remove (uninstall) the Oracle Password Filter for Microsoft Active Directory as described in this section.

To remove the Oracle Password Filter for Microsoft Active Directory:

1. Open in a text editor the **prepAD.Idif** 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.Idif 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 back-end 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 `deleteBackendSchema.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: ( 2.16.840.1.113894.8.1.1016 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.

# Integrating with Oracle Directory Server Enterprise Edition (Connected Directory)

This chapter outlines the procedures for integrating Oracle Identity Management with Oracle Directory Server Enterprise Edition connected directory (previously known as Sun Java System Directory Server, and, before that, SunONE iPlanet).

## 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 the following chapters:

- [Introduction to Oracle Directory Integration Platform](#)
- [Managing the Oracle Directory Integration Platform](#)
- [Understanding the Oracle Directory Synchronization Service](#)
- [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 11g Release 1 (11.1.1), available on Oracle Technology Network at <http://www.oracle.com/technology/>

## 23.1 Verifying Synchronization Requirements for Oracle Directory Server Enterprise Edition

Before configuring basic or advanced synchronization with Oracle Directory Server Enterprise Edition, ensure that your environment meets the necessary synchronization requirements by following the instructions in "[Verifying Synchronization Requirements](#)". Before synchronizing with Oracle Directory Server Enterprise Edition, you must also perform the following steps:

1. Enable change logging on Oracle Directory Server Enterprise Edition.
2. Enable the retro change log plug-in, as described in [Enabling the Retro Change Log for Oracle Directory Server Enterprise Edition](#)

3. Configure the retro change log to record specified attributes of an entry that is deleted, as described in "[To Configure the Retro Change Log to Record Attributes of a Deleted Entry](#)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.
4. Add an attribute to an existing list of specified attributes, by running the following command:

```
$ dsconf set-server-prop -w /tmp/pwd -h host -p port retro-cl-deleted-entry-attr+:attribute
```

For more information, see [Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition](#).

## 23.2 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.

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, see "[Creating Import and Export Synchronization Profiles Using `expressSyncSetup`](#)".

## 23.3 Configuring Advanced Integration with Oracle Directory Server Enterprise Edition

You can also use the `expressSyncSetup` command to create Oracle Enterprise Manager Fusion Middleware Control additional synchronization profiles from the templates.

The import and export synchronization profiles created 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 these synchronization profiles are created using predefined assumptions, you must further customize them for your environment by performing the following steps:



### Note:

When you install Oracle Directory Integration Platform, import and export template files are automatically created (`ORACLE_HOME/ldap/odi/conf/`). The template files 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



- [Understanding How to Plan Integration with Oracle Directory Server Enterprise Edition](#)
- [Configure the Realm for Oracle Directory Server Enterprise Edition](#)
- [Understanding How to Customize the ACLs for Oracle Directory Server Enterprise Edition](#)
- [Customize Attribute Mappings for Oracle Directory Server Enterprise Edition](#)
- [About How to Customize the Oracle Directory Server Enterprise Edition Connector to Synchronize Deletions](#)
- [Understanding How to Synchronize Passwords for Oracle Directory Server Enterprise Edition](#)
- [Synchronizing in SSL Mode](#)
- [Perform Post-Configuration and Administrative Tasks](#)

### 23.3.1 Understanding How to Plan Integration with Oracle Directory Server Enterprise Edition

Plan your integration by reading [Connected Directory Integration Concepts and Considerations](#), particularly "Oracle Directory Server Enterprise Edition (Sun Java System Directory Server) Integration Concepts". 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](#)".

### 23.3.2 Configure the Realm for Oracle Directory Server Enterprise Edition

Configure the realm by following the instructions in [Configuring the Realm](#).

### 23.3.3 Understanding How to Customize the ACLs for Oracle Directory Server Enterprise Edition

Customize ACLs as described in [Customizing Access Control Lists](#).

### 23.3.4 Customize Attribute Mappings for Oracle Directory Server Enterprise Edition

When integrating with Oracle Directory Server Enterprise Edition, the following attribute-level mapping is mandatory for all objects:

```
Targetdn:1: :person:orclsourceobjectdn: : orclSUNOneobject:
```

**Example 23-1 Attribute-Level Mapping for the User Object in Oracle Directory Server Enterprise Edition**

```
Cn:1: :person: cn: :person:  
sn:1: :person: sn: :person:
```

### Example 23-2 Attribute-Level Mapping for the Group Object in Oracle Directory Server Enterprise Edition

```
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](#).

## 23.3.5 About How to Customize the Oracle Directory Server Enterprise Edition Connector to Synchronize Deletions

If you want to synchronize deletions, and the mapping rules have mandatory attributes, then ensure that they are present in the change log when the entry is deleted. You must add `Objectclass` and other values to the list of attributes to be included when an entry is deleted, as described in "To Configure the Retro Change Log to Record Attributes of a Deleted Entry" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.

## 23.3.6 Understanding How to Synchronize Passwords for Oracle Directory Server Enterprise Edition

You can synchronize the password, as described in [Password Synchronization](#).

## 23.3.7 Synchronizing in SSL Mode

You must configure Oracle Directory Server Enterprise Edition for synchronization in SSL mode.

To do so, by following the instructions in [Configuring the Connected Directory Connector for Synchronization in SSL Mode](#).



### Note:

Oracle recommends you to synchronize the password using the SSL communication for the back-end directory and connected directory.

## 23.3.8 Perform Post-Configuration and Administrative Tasks

This section describes the task you must complete after configuring advanced integration with Oracle Directory Server Enterprise Edition.

See [Managing Integration with a Connected Directory](#) for information on post-configuration and ongoing administration tasks.

# Integrating with IBM Tivoli Directory Server

This chapter outlines the procedures for integrating Oracle Identity Management with IBM Tivoli Directory Server.

## 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:

- [Introduction to Oracle Directory Integration Platform](#)
- [Managing the Oracle Directory Integration Platform](#)
- [Understanding the Oracle Directory Synchronization Service](#)
- [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 11gRelease 1 (11.1.1), available on Oracle Technology Network at <http://www.oracle.com/technology/>

## 24.1 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.

You must complete the steps described in "[Verifying Synchronization Requirements](#)". Before synchronizing with IBM Tivoli Directory Server, you must also perform the following steps:

1. 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 tombstones.
2. Enable change logging on IBM Tivoli Directory Server.

## 24.2 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`](#)".

## 24.3 Configuring Advanced Integration with IBM Tivoli Directory Server

You can also use the `expressSyncSetup` command to create additional synchronization profiles from the template files.

The import and export synchronization profiles that you created 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 these synchronization profiles are created using predefined assumptions, you must further customize them for your environment by performing the following steps:

### Note:

When you install Oracle Directory Integration Platform, import and export template files (`ORACLE_HOME/ldap/odi/conf`) are automatically created for each of the supported third-party directories. You can use the template files, to create the user's profile. The sample files 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

- [Understanding How to Plan Integration with IBM Tivoli Directory](#)
- [Configure the Realm for IBM Tivoli Directory](#)
- [Understanding How to Customize the ACLs for IBM Tivoli Directory](#)
- [Customize Attribute Mappings for IBM Tivoli Directory](#)
- [Customizing the IBM Tivoli Directory Server Connector to Synchronize Deletions](#)
- [Synchronize Passwords for IBM Tivoli Directory](#)

- [Understand How to Synchronize IBM Tivoli Directory in SSL Mode](#)
- [Configuring the IBM Tivoli Directory Server External Authentication Plug-in](#)
- [Perform Post-Configuration and Administrative Tasks](#)

## 24.3.1 Understanding How to Plan Integration with IBM Tivoli Directory

Plan your integration by reading [Connected Directory Integration Concepts and Considerations](#), particularly "IBM Tivoli Directory Server Integration Concepts". 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](#)".

## 24.3.2 Configure the Realm for IBM Tivoli Directory

Configure the realm by following the instructions in "[Configuring the Realm](#)".

## 24.3.3 Understanding How to Customize the ACLs for IBM Tivoli Directory

Customize ACLs as described in "[Customizing Access Control Lists](#)".

## 24.3.4 Customize Attribute Mappings for IBM Tivoli Directory

When integrating with IBM Tivoli Directory Server, the following attribute-level mapping is mandatory for all objects:

```
targetdn: : :top:orclSourceObjectDN: :orclTDSObject:
```

### **Example 24-1 Attribute-Level Mapping for the User Object in IBM Tivoli Directory Server**

```
Cn:1: :person: cn: :person:  
sn: : :person: sn: :person:
```

### **Example 24-2 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](#)".

## 24.3.5 Customizing 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.

### See Also:

IBM Tivoli Directory Server documentation for details about configuring tombstones.

## 24.3.6 Synchronize Passwords for IBM Tivoli Directory

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.

## 24.3.7 Understand How to Synchronize IBM Tivoli Directory 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](#)".

## 24.3.8 Configuring 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=subconfigsentry
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=orcluser2))
orclpluginflexfield;host2: myhost.us.example.com
orclpluginflexfield;port2: 636
orclpluginflexfield;isssl2: 1
orclpluginflexfield;host: myhost.us.example.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=subconfigsentry
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: myhost.us.example.com
orclpluginflexfield;port2: 636
orclpluginflexfield;isssl2: 1
orclpluginflexfield;host: myhost.us.example.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](#)" to configure the plug-in.

## 24.3.9 Perform Post-Configuration and Administrative Tasks

This section describes the task you must complete after configuring advanced integration with IBM Tivoli Directory Server.

Read [Managing Integration with a Connected Directory](#) for information on post-configuration and ongoing administration tasks.



# 25

## 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.

### Topics:

- [Verify Synchronization Requirements for Novell eDirectory or OpenLDAP](#)
- [Configuring Basic Synchronization with Novell eDirectory or OpenLDAP](#)
- [Configuring Advanced Integration with Novell eDirectory or OpenLDAP](#)

### Note:

Before continuing with this chapter, you should be familiar with the concepts presented in previous chapters. The following chapters in particular are important:

- [Introduction to Oracle Directory Integration Platform](#)
- [Managing the Oracle Directory Integration Platform](#)
- [Understanding the Oracle Directory Synchronization Service](#)
- [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.

### 25.1 Verify 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.

You must complete the steps described in "[Verifying Synchronization Requirements](#)".

 **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.

## 25.2 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`](#)".

## 25.3 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:

1. Modify the mapping rule so each profile creates the user in a different container. Refer to "[Customizing Mapping Rules](#)" for more information.
2. Modify the reconciliation rules in the mapping file to synchronize only a specific subset of users. Refer to "[About How to Define a Reconciliation Rule?](#)" for more information.

## 25.4 Configuring Advanced Integration with Novell eDirectory or OpenLDAP

You can use the `expressSyncSetup` command or Oracle Enterprise Manager Fusion Middleware Control to create additional synchronization profiles from the templates.

The import and export synchronization profiles created 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 these synchronization profiles are created using predefined assumptions, you must further customize them for your environment by performing the following steps:

 **Note:**

When you install Oracle Directory Integration Platform, import and export template files (`ORACLE_HOME/ldap/odi/conf`) are created for each of the supported directories. The template files 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 synchronization profiles templates 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.

- [Understanding How to Plan Integration with Novell eDirectory or OpenLDAP](#)
- [About Realm Configuration for Novell eDirectory or OpenLDAP](#)
- [Customize the Search Filter to Retrieve Information from Novell eDirectory or OpenLDAP](#)
- [Understanding the ACLs Customization for Novell eDirectory or OpenLDAP](#)
- [Customize Attribute Mappings for Novell eDirectory](#)
- [Customizing the Novell eDirectory or OpenLDAP Connector to Synchronize Deletions](#)
- [Synchronization Parameters for the Advanced Configuration Information Attribute](#)
- [Configuring the OpenLDAP Connector to Synchronize Passwords](#)
- [Synchronize the Novell eDirectory or OpenLDAP connector in SSL Mode](#)
- [Configure the Novell eDirectory or OpenLDAP External Authentication Plug-in](#)
- [Perform Post-Configuration and Administrative Tasks](#)

## 25.4.1 Understanding How to Plan Integration with Novell eDirectory or OpenLDAP

Plan your integration by reading [Connected Directory Integration Concepts and Considerations](#), particularly "[Novell eDirectory and OpenLDAP Integration Concepts](#)". Be sure to create a new profile by copying the existing eDirectory or OpenLDAP template profile by following the instructions in "[Creating Synchronization Profiles](#)".

## 25.4.2 About Realm Configuration for Novell eDirectory or OpenLDAP

Configure the realm by following the instructions in "[Configuring the Realm](#)".

## 25.4.3 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))
```

### Note:

- 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](#)" for more information about the `manageSyncProfiles` command.

### See Also:

The appendix on the LDAP filter definition in *Oracle Internet Directory* for instructions on configuring an LDAP search filter.

## 25.4.4 Understanding the ACLs Customization for Novell eDirectory or OpenLDAP

Customize ACLs as described in "[Customizing Access Control Lists](#)".

## 25.4.5 Customize Attribute Mappings for Novell eDirectory

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 25-1 Attribute-Level Mapping for the User Object in Novell eDirectory or OpenLDAP

```
Cn:1 : :person: cn: :person:
sn:1 : :person: sn: :person:
```

### Example 25-2 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](#)".

## 25.4.6 Customizing 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 "[About Synchronization from Novell eDirectory or OpenLDAP to the Oracle Back-end Directory](#)". 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 `manageSyncProfiles` 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:*
```

 **Note:**

For more information, see:

- [About How to Define a Reconciliation Rule?](#)
- [Reconciliation Rules Used to Synchronize Deletions?](#)

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.

## 25.4.7 About How to 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.

## 25.4.8 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))

## 25.4.9 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 25-1](#) to specify additional configuration information.

### Tip:

Refer to the **Advanced** tab section in “[Creating Synchronization Profiles](#)” for a description of all Advanced Configuration parameters for synchronization profiles.

**Table 25-1 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.

**Table 25-1 (Cont.) Novell eDirectory and OpenLDAP Synchronization Parameters for the Advanced Configuration Information Attribute**

Parameter	Description
SearchTimeDeltaInSeconds	<p>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.</p> <p>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.</p> <p>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.</p>
CheckAllEntries	<p>Determines how deleted entries in Novell eDirectory or OpenLDAP are synchronized with the Oracle back-end directory. If you assign a value of <code>true</code> 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 <code>false</code> 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 <code>false</code> to this parameter.</p>



**Table 25-1 (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. If one computer's clock is drifting apart from the other, then must periodically update this value, by keeping the clocks synchronized using the Network Time Protocol (NTP).
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.

## 25.4.10 Configuring 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.

### 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 Internet Directory.

 **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 `orclPwPolicyEnable` and `orclPwEncryptionEnable` attributes in the entry `cn=PwPolicyEntry,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=PwPolicyEntry,cn=common,cn=products,cn=oraclecontext,DN_of_realm.
changetype: modify
replace: orclpwpolicyenable
orclpwpolicyenable: 1
-
replace: orclpwdencryptionenable
orclpwdencryptionenable: 1
```

 **See Also:**

- ["Configuring the Connected Directory Connector for Synchronization in SSL Mode"](#)
- The section ["Configuring Mapping Rules"](#) 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 Internet Directory* for information about enabling reversible encryption.

## 25.4.11 Synchronize the Novell eDirectory or OpenLDAP connector 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"](#).

## 25.4.12 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"](#).

## 25.4.13 Perform Post-Configuration and Administrative Tasks

This section describes the task you must complete after configuring advanced integration with Novell eDirectory or OpenLDAP.

See [Managing Integration with a Connected Directory](#) for information on post-configuration and ongoing administration tasks.

# 26

## Managing Integration with a Connected Directory

Understand the post-configuration and ongoing administration tasks after integration with a connected directory.

**Topics:**

- [Performing Tasks After Configuring with a Connected Directory](#)
- [Typical Management of Integration with a Connected Directory](#)

### 26.1 Performing Tasks After Configuring with a Connected Directory

This section describes the task you must complete after configuring with a connected directory.

Complete the following:

1. Migrate data from one directory to the other as needed. This is described in "[Bootstrapping Data Between Directories](#)".
2. Use the `activate` operation of the `manageSyncProfiles` command to enable the synchronization profile by entering the following command:

```
manageSyncProfiles activate -profile profile_name
```

You can also use Oracle Enterprise Manager Fusion Middleware Control to enable the synchronization profile, as described in [Enabling and Disabling Synchronization Profiles](#).

### 26.2 Typical Management of Integration with a Connected Directory

Know more about the management tasks, after integration with a Connected Directory.

**Topics:**

- [Overview of Management Task for after Integration with a Connected Directory](#)
- [Bootstrapping Data Between Directories](#)
- [Managing a Third-Party Directory External Authentication Plug-in](#)
- [Granting Password Privilege for Oracle Unified Directory Acting As Connected Directory](#)

## 26.2.1 Overview of Management Task for after 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.

## 26.2.2 Bootstrapping Data Between Directories

Bootstrapping is sometimes called data migration. You must bootstrap data after the third-party directory connector and plug-in configurations are complete.

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 [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.

## 26.2.3 Managing a Third-Party Directory External Authentication Plug-in

This topic explains how to delete, disable, and re-enable a third-party external authentication plug-in.

### Topics:

- [Deleting a Third-Party Directory External Authentication Plug-in](#)
- [Disabling a Third-Party External Authentication Plug-in](#)
- [Re-enabling a Third-Party External Authentication Plug-in](#)

Oracle Unified Directory and Oracle Directory Server Enterprise Edition back-end directories uses pass-through authentication for passing authentication through to a connected directory like Microsoft Active Directory for users coming from Oracle Unified Directory or Oracle Directory Server Enterprise Edition. For more information, see:

- The section "Understanding Pass-Through Authentication" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Unified Directory*.
- The section "[Pass-Through Authentication](#)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Directory Server Enterprise Edition*.

### 26.2.3.1 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=subconfigsentry"
```

```
ldapdelete -h host -p port -D binddn -q \  
"cn=adwhenbind,cn=plugin,cn=subconfigsentry"
```

### 26.2.3.2 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=subconfigsentry  
changetype: modify  
replace: orclpluginenable  
orclpluginenable: 0
```

```
dn: cn=adwhenbind,cn=plugin,cn=subconfigsentry  
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.

### 26.2.3.3 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.

### 26.2.4 Granting Password Privilege for Oracle Unified Directory Acting As Connected Directory

Use the `ldapmodify` command to grant the password-reset privilege for Oracle Unified Directory (Connected Directory).

If Oracle Unified Directory is the connected directory then you must grant the password-reset privilege to the export profiles by running the following command:

```
$ ldapmodify -h localhost -p 1389 -D "cn=Directory Manager" -w password
dn: <your user DN>
changetype: modify
add: ds-privilege-name
ds-privilege-name: password-reset
```

# Part VII

## Appendixes

You may need to consult this supplemental information, such as the example profile properties file with the `manageSyncProfiles` command, case study for Oracle Directory Integration Platform deployment, starting and stopping the Oracle Stack and troubleshooting information.

This part contains the following appendixes:

- [Example Properties File for Synchronization Profiles](#)
- [Case Study: A Deployment of Oracle Directory Integration Platform](#)
- [Starting and Stopping the Oracle Stack](#)
- [Troubleshooting the Oracle Directory Integration Platform](#)



# A

## Example Properties File for Synchronization Profiles

You can use an example profile properties file with the `manageSyncProfiles` command and its `-f` option 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"](#) for more information about `manageSyncProfiles`.

**Topic:**

- [Example Properties File for Synchronization Profiles](#)

### A.1 Example Profile Properties File

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 ##  
## Oracle Back-end  
Directory. #####  
#####  
  
# 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 Oracle Back-end  
Directory 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
# completed then the next execution will not resume until 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.agentexeccommand =

# Connected Directory Url : The 3rd party directory location
# The property is of the 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 Oracle Back-end Directory
#
# 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 Oracle Back-end Directory attribute : Specifies the matching
attribute
# on Oracle Back-end Directory for import from the connected directory
#
odip.profile.backendfilter = orclObjectGUID

# Change Number : Specifies the last applied change number. In case of
# an export profile this number refer's to Oracle Back-end Directory'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, Oracle Back-end Directory, 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 default value.
#
odip.profile.updateChangeNumberatCreate = TRUE
```

# B

## 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.

### Topics:

- [Components in the MyCompany Enterprise](#)
- [Requirements of the MyCompany Enterprise](#)
- [Overall Deployment in the MyCompany Enterprise](#)
- [User Creation and Provisioning in the MyCompany Enterprise](#)
- [Modification of User Properties in the MyCompany Enterprise](#)
- [Deletion of Users in the MyCompany Enterprise](#)

### B.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

### B.2 Requirements of the MyCompany Enterprise

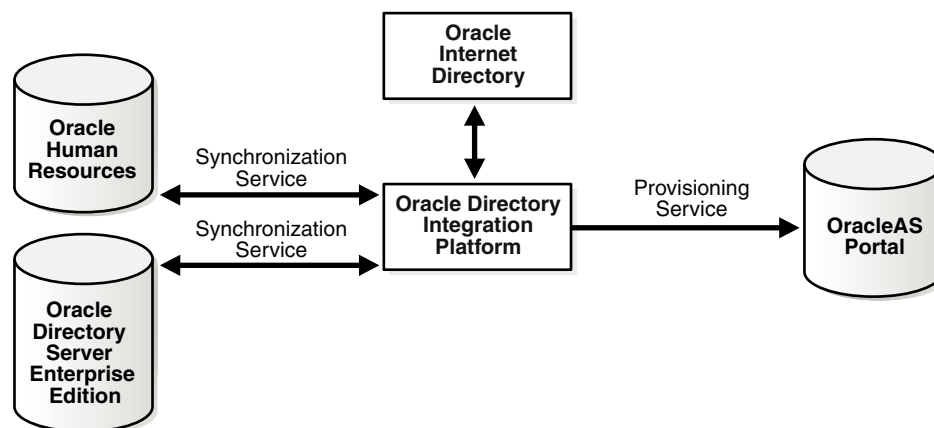
The MyCompany enterprise requires that:

- 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, 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.

### B.3 Overall Deployment in the MyCompany Enterprise

[Figure B-1](#) illustrates the various components and their relationships to each other.

**Figure B-1 Example of Oracle Directory Integration Platform in the MyCompany Deployment**



In the example in [Figure B-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.

## B.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 B-2](#) illustrates how Oracle Directory Integration Platform performs this task.

**Figure B-2 User Creation and Provisioning**

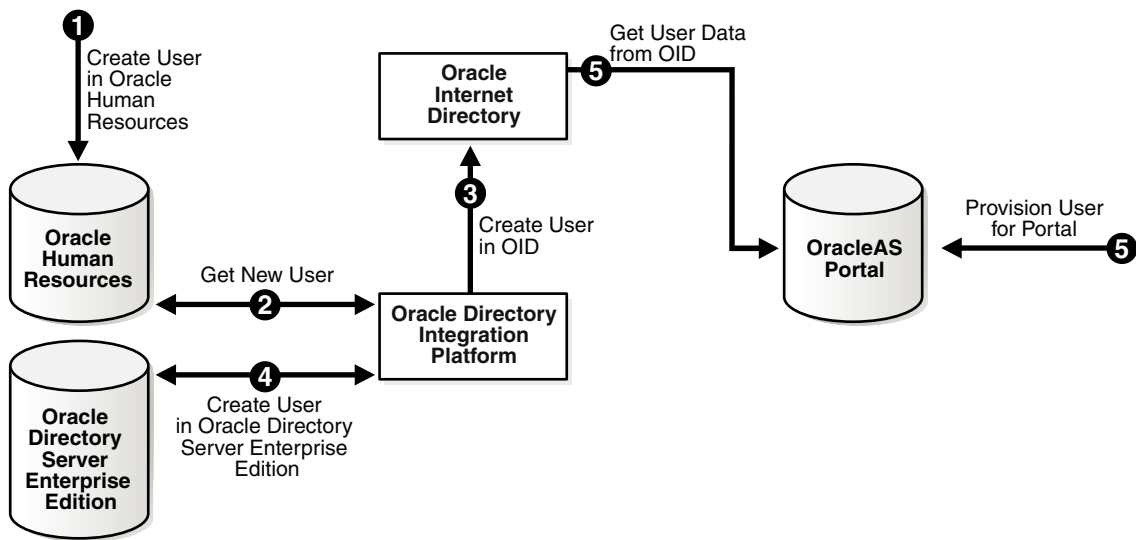


Figure B-2 shows the creation of a new user in Oracle Human Resources, which, in turn, causes an entry for that user to be created in Oracle Internet Directory and Oracle Directory Server Enterprise Edition (previously Sun Java System Directory Server). It also shows the process of provisioning the user 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.

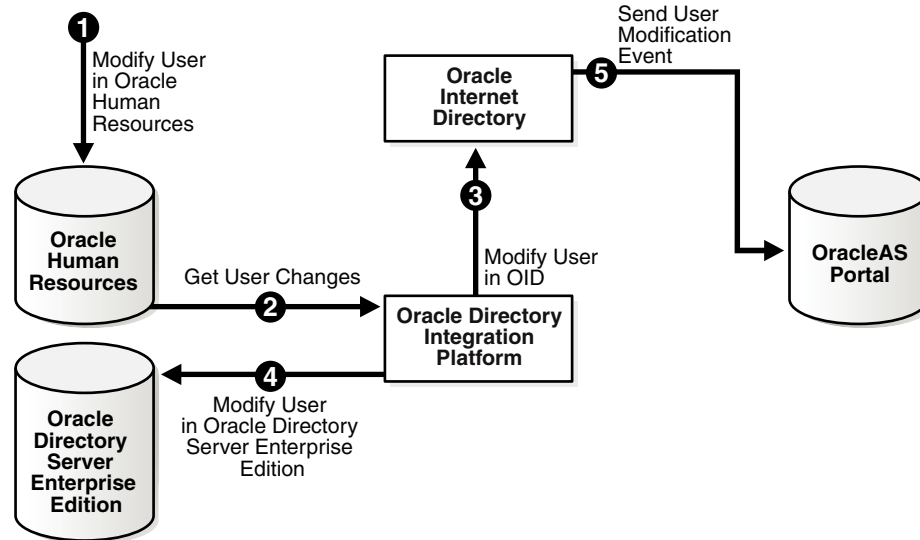
Note that Oracle Directory Integration Platform does not directly notify Oracle Portal about new users. This is because not all users created in Oracle Human Resources need access to all services. In this case, the deployment must explicitly provision the users to use these services, as in Step 5.

## B.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 B-3](#)

illustrates the actions that Oracle Directory Integration Platform takes to meet this requirement.

**Figure B-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.

## B.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 B-4 shows the flow of events during the deletion of users.



**Figure B-4 Deletion of Users from the Corporate Human Resources**

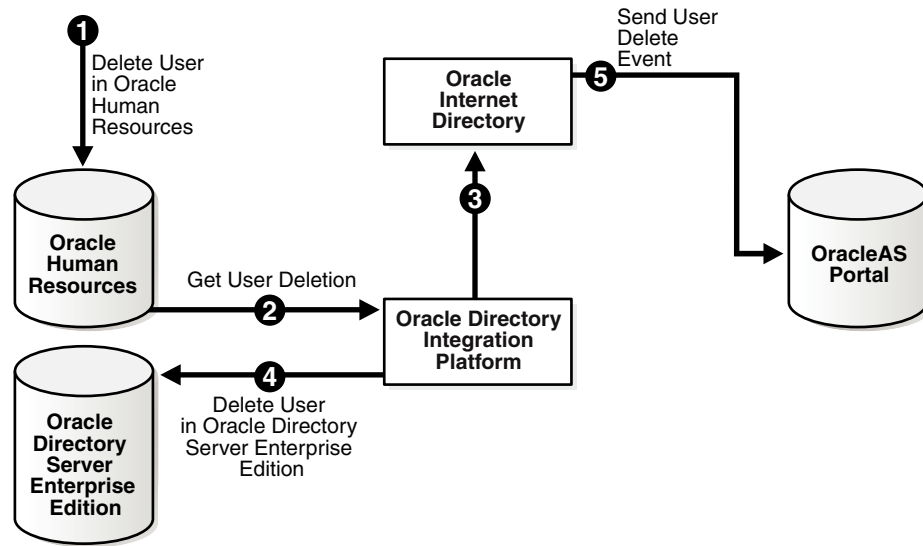


Figure B-4 shows the process by which Oracle Directory Integration Platform communicates the deletion of users to all systems in the enterprise. The process is as follows:

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.

# C

## 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.

### Topics:

- [Starting the Stack](#)
- [Stopping the Stack](#)
- [About How to Start or Stop the Oracle Unified Directory Server](#)
- [About How to Start or Stop the Oracle Internet Directory Server](#)
- [About How to Start, Stop, or Restart a Oracle Directory Server Enterprise Edition Instance](#)

### C.1 Starting the Stack

Use the steps in the following order to start the stack components.

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.

(Unix)

```
DOMAIN_NAME/bin/startWebLogic.sh
```

(Windows)

```
DOMAIN_NAME\bin\startWebLogic.cmd
```

#### Note:

Where `DOMAIN_NAME` is the root directory of the domain. (The name of this directory is the name of the domain.). By default, this directory is `ORACLE_HOME\user_projects\domains\DOMAIN_NAME`.

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.

(Unix)

```
DOMAIN_NAME/bin/startNodeManager.sh
```

(Windows)

```
DOMAIN_NAME\bin\startNodeManager.cmd
```

4. Start the Oracle WebLogic managed components, such as Oracle Directory Integration Platform and Oracle Directory Services Manager.

(Unix)

```
DOMAIN_NAME/bin/startManagedWebLogic.sh managed_server_name admin_url
```

(Windows)

```
DOMAIN_NAME\bin\startManagedWebLogic.cmd managed_server_name admin_url
```

Where `managed_server_name` specifies the name of the Managed Server (The default value is `wls_ods1`.) and `admin_url` specifies the listen address (host name, IP address, or DNS name) and port number of the domain's Administration Server.

Example

```
c:\Oracle\Middleware\user_projects\domains\base_domain\bin\startManagedWebLogic.cmd wls_ods1 http://AdminHost:7001
```

To start managed components in the background, you can use the Oracle WebLogic Administration Console. See *Using the WebLogic Server Administration Console* in *Understanding Oracle WebLogic Server*.

You can view the status of Oracle WebLogic managed components with Fusion Middleware Control. See *Using Fusion Middleware Control* in *Understanding Oracle WebLogic Server*.

## C.2 Stopping the Stack

Use the steps in the following order to stop the stack components from the command line.

You can stop the Administration Server and all the managed servers by using Oracle WebLogic Administration Console. See *Using the WebLogic Server Administration Console* in *Understanding Oracle WebLogic Server*.

1. Stop the Oracle WebLogic managed components, such as Oracle Directory Integration Platform and Oracle Directory Services Manager.

(Unix)

```
DOMAIN_NAME/bin/stopManagedWebLogic.sh managed_server_name admin_url
```

(Windows)

```
DOMAIN_NAME\bin\stoptManagedWebLogic.cmd managed_server_name admin_url
```

Where `DOMAIN_NAME` is the root directory of the domain, `managed_server_name` specifies the name of the Managed Server (The default value is `wls_ods1`.), and `admin_url` specifies the listen address (host name, IP address, or DNS name) and port number of the domain's Administration Server.

Example

```
c:\Oracle\Middleware\user_projects\domains\base_domain\bin\stopManagedWebLogic.cmd wls_ods1 http://AdminHost:7001
```

2. Stop the WebLogic Administration Server.

(Unix)

```
DOMAIN_NAME/bin/stopWebLogic.sh
```

(Windows)

```
DOMAIN_NAME\bin\stopWebLogic.cmd
```

 **Note:**

Where `DOMAIN_NAME` is the directory in which you located the domain. By default, this directory is `ORACLE_HOME\user_projects\domains\DOMAIN_NAME`.

3. If you want to stop the Node Manager, you can use the `stopNodeManager.sh` command.

(Unix)

```
DOMAIN_NAME/bin/stopNodeManager.sh
```

(Windows)

```
DOMAIN_NAME\bin\stopNodeManager.cmd
```

4. 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
```

## C.3 About How to Start or Stop the Oracle Unified Directory Server

Use the `start-ds` or `stop-ds` commands to start or stop the Oracle Unified Directory Server.

See Starting and Stopping the Server in *Administering Oracle Unified Directory*.

## C.4 About How to Start or Stop the Oracle Internet Directory Server

Use the `start()` or `shutdown()` commands to start or stop the Oracle Internet Directory Server.

See Starting the Oracle Internet Directory Server by Using WLST Command — start() or Stopping the Oracle Internet Directory Server by Using WLST Command — shutdown() in *Oracle Fusion Middleware Administering Oracle Internet Directory*.

## C.5 About How to Start, Stop, or Restart a Oracle Directory Server Enterprise Edition Instance

Use the `dsadm` command to start, stop, or restart a Oracle Directory Server Enterprise Edition Instance.

See [Starting, Stopping, and Restarting a Directory Server Instance](#) in *Administrator's Guide for Oracle Directory Server Enterprise Edition*.

# D

## 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.

### Topics:

- [Checklist for Troubleshooting Oracle Directory Integration Platform](#)
- [General Issues](#)
- [Configuration Issues](#)
- [Problems and Solutions](#)
- [Troubleshooting Synchronization](#)
- [Troubleshooting Integration with Microsoft Active Directory](#)
- [Troubleshooting SSL/TLS](#)
- [Need More Help?](#)

### 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*

## D.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 Oracle 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.

 **Note:**

- 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.
- For more information, see "[dipStatus Utility](#)".

- 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.

## D.2 General Issues

This section describes issues that affect Oracle Directory Integration Platform.

### Topics:

- [LDIF Files That Contain Non-ASCII Characters Will Cause the `testProfile` Command Option to Fail if the LDIF File has Native Encoding](#)
- [Some Changes May Not Get Synchronized Due to Race Condition in Heavily-Loaded Source Directory](#)
- [Synchronization Continues After Stopping Oracle Directory Integration Platform](#)

- [Synchronization Continues After Disabling Synchronization Profiles for Oracle Unified Directory \(Back-End Directory\)](#)
- [Synchronization of Deleted Objects Fails](#)

## D.2.1 LDIF Files That Contain Non-ASCII Characters Will Cause the `testProfile` Command Option to Fail if the LDIF File has Native Encoding

This section describes how to resolve `testProfile` Command Option failure.

When running DIP Tester from a command-line, the `manageSyncProfiles testProfile` command will fail if the `-ldiffile` option is specified and the LDIF file contains non-ASCII characters.

Note that LDIF files with UTF-8 encoding are not impacted by this limitation. If an LDIF file containing multibyte characters cannot be saved with UTF-8 encoding, then use the following workaround:

1. From a command-line, add the entry using the `ldapadd` command and include the `-E` option to specify the locale. See the *Oracle Fusion Middleware User Reference for Oracle Identity Management* for the required command syntax.
2. Get the specific `changeNumber` for the last add operation.
3. Execute the `testProfile` command using the `changeNumber` from the previous step.

For more information, see [Running DIP Tester From the WLST Command-Line Interface](#).

## D.2.2 Some Changes May Not Get Synchronized Due to Race Condition in Heavily-Loaded Source Directory

If the source directory is heavily-loaded, a race condition may occur where database commits cannot keep pace with updates to the `lastchangenumber`. If this race condition occurs, Oracle Directory Integration Platform may not be able to synchronize some of the changes.

To work around this issue, perform the following steps to enable database commits to keep pace with the `lastchangenumber`:

- Increase the value of the synchronization profile's Scheduling Interval.
- Control the number of times the search is performed on the source directory during a synchronization cycle by setting the `searchDeltaSize` parameter in the profile. Oracle suggests starting with a value of 10, then adjusting the value as needed.



## D.2.3 Synchronization Continues After Stopping Oracle Directory Integration Platform

If you stop the Oracle Directory Integration Platform application during synchronization, the synchronization process that the Quartz scheduler started will continue to run.

To work around this issue, restart the Oracle WebLogic Managed Server hosting Oracle Directory Integration Platform or redeploy the Oracle Directory Integration Platform application.

## D.2.4 Synchronization Continues After Disabling Synchronization Profiles for Oracle Unified Directory (Back-End Directory)

If you disable an existing synchronization profile for Oracle Unified Directory (back-end directory), the synchronization process will continue to run.

To work around this issue, perform one of the following steps:

- Restart the Oracle WebLogic Managed Server hosting Oracle Directory Integration Platform. For more information about restarting Oracle WebLogic Managed Server, see [Starting the Stack](#).
- Enable the External Change Log for `cn=Products,cn=oraclecontext` using the `dsreplication` command on the command line:

```
dsreplication enable-changelog -h localhost -p 4444 -  
D "cn=directory manager" -j pwd-file -r 8989 -b  
cn=Products,cn=oraclecontext -X -n
```

## D.2.5 Synchronization of Deleted Objects Fails

This section explains how to resolve issue if the deleted objects are not synchronized.

If the deleted objects are not synchronized and the domain mapping rule is as shown below:

```
dc=example,dc=com:cn=users,dc=example,dc=com:cn=*,cn=users,dc=example,dc=c  
om
```

To synchronize the deleted objects, you must remove the following line from the domain mapping rule:

```
cn=*,cn=users,dc=example,dc=com
```

If the `cn=*` is mandatory because the RDN is different between the source and the destination, then the attribute used as the RDN must be marked as required in the attribute mapping rule.

## D.3 Configuration Issues

This section describes configuration issues that affect Oracle Directory Integration Platform.

**Topics:**

- [You may Need to Restart the Directory Integration Platform After Running dipConfigurator Against Oracle Unified Directory](#)
- [When Configuring a Profile, you may Need to Scroll Past a Section of Whitespace to View Mapping Rules](#)

## D.3.1 You may Need to Restart the Directory Integration Platform After Running dipConfigurator Against Oracle Unified Directory

After running `dipConfigurator` utility against an Oracle Unified Directory endpoint, if you are unable to open the Oracle Directory Integration Platform UI in Enterprise Manger, stop and start Oracle Directory Integration Platform to fix the UI problem.

## D.3.2 When Configuring a Profile, you may Need to Scroll Past a Section of Whitespace to View Mapping Rules

If you are using Internet Explorer to view the Directory Integration Platform (DIP) UI, you may need to scroll past a large blank space to see the profile mapping rules section. This issue is not known to affect other browsers.

## D.4 Problems and Solutions

This section describes common problems and solutions for Oracle Directory Integration Platform.

**Topics:**

- [Provisioning Errors and Problems](#)
- [Synchronization 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](#)".

### D.4.1 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. See *Managing Garbage Collection in Administering 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 directory server administration in *Administering Oracle Internet Directory*. This topic 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:

```
(UNIX) DOMAIN_HOME/servers/server_name/logs  
(Windows) DOMAIN_HOME\servers\server_name\logs
```

The default name of the log file is `server-name-diagnostic.log`.

#### Note:

### Problem

Provisioning Profiles Not Getting Executed by the DIP Provisioning Server.

### Solution

Using Oracle Enterprise Manager Fusion Middleware Control or the `manageProvProfiles` 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:

```
(UNIX) DOMAIN_HOME/servers/server_name/logs  
(Windows) DOMAIN_HOME\servers\server_name\logs
```

The default name of the log file is `server-name-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)
```

## D.4.2 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://support.oracle.com/>.

## Problem

[DIP-10247] - Not able to construct DN

## Solution

This error normally happens, if you define the domain mapping rule using a `%` wildcard:

```
cn=Users,dc=example,dc=com : ou=employees,dc=example,dc=com :
uid=%,ou=employees,dc=example,dc=com
```

To resolve the issue, set the `required` flag in the in the domain mapping rule as follows:

```
SAMAccountName : 1 : : user : uid : : inetorgperson :
```

## Problem

Hashed Password Synchronization to Oracle Unified Directory Fails

## Solution

Before synchronizing hashed passwords (By copying the `userPassword` attribute from any server to the Oracle Unified Directory), you must run the following command to allow hashed passwords storing:

```
dsconfig set-password-policy-prop --policy-name Default\ Password\ Policy --set
allow-pre-encoded-passwords:true
```

## 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://support.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**

Invalid value is specified for the parameter "odip.bootstrap.srctype". Please verify: DB

**Solution**

This problem occurs when you are using a Oracle Database with the back-end directory and the Oracle Database is the source. To resolve the problem, do the following:

 **Note:**

If a connected Oracle Database is used as the source directory then bootstrapping is not supported.

1. Enable the existing synchronization profile using Oracle Enterprise Manager Fusion Middleware Control. See [Enabling and Disabling Synchronization Profiles](#).
2. Log in to the Oracle Enterprise Manager Fusion Middleware Control and specify 0 as the value for the **Last Change Number** in the **Advanced** tab of the Edit Synchronization Profile page:  
See [Editing Synchronization Profiles](#).
3. Run DIP Tester using the Enterprise Manager. See [Running DIP Tester From the Enterprise Manager User Interface](#).
4. Go to the **Additional Configuration Parameters** section. For **Check All Entries**, specify `true` as the value.

**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://support.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](#)" for more information about the `manageSyncProfiles` command.

## D.4.3 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 or 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.

## D.4.4 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, schemaNamingContext` 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](#)".

### 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](#)".

### 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](#)".

**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](#)".

**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](#)".

**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](#)".

**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](#)".

**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](#)".

## D.5 Troubleshooting Synchronization

This section describes how to troubleshoot synchronization with Oracle Directory Integration Platform.

**Topics:**

- [Oracle Directory Integration Platform Synchronization Process Flow](#)
- [Understanding Synchronization Profile Registration](#)
- [Understanding the diagnostic.log File](#)

## D.5.1 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.

### Topics:

- [Oracle Directory Integration Platform Synchronization Process Flow for an Import Profile](#)
- [Oracle Directory Integration Platform Synchronization Process Flow for an Export Profile](#)

### D.5.1.1 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.

## D.5.1.2 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.

## D.5.2 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`](#)" for more information.

## D.5.3 Understanding the diagnostic.log File

This section explains how to understand the Oracle Directory Integration Platform `diagnostic.log` file.

The Oracle Directory Integration Platform `diagnostic.log` file is located at the following location:

```
(UNIX) DOMAIN_HOME/servers/server_name/logs
(Windows) DOMAIN_HOME\servers\server_name\logs
```

The default name of the log file is `server-name-diagnostic.log`.

### Note:

The file name is `NAME_OF_MANAGED_SERVER-diagnostic.log`.

The 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.

### Note:

Oracle Unified Directory and Oracle Directory Server Enterprise Edition only support `SSL mode 2`.

- **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.

```
[wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid:
[ACTIVE].ExecuteThread: '1' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP]
```

```
[wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid:
[ACTIVE].ExecuteThread: '1' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP] SSL Mode : 1
```



```

[wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid:
[ACTIVE].ExecuteThread: '1' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP] Host: HOST_NAME: PORT

[wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid: Scheduler] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP] Scheduler intialized

[wls_ods1] [NOTIFICATION] [DIP-10571] [oracle.dip] [tid: Scheduler]
[userId: <anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP] Connection to LDAP
Server Successful

[wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid: Scheduler] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP]
OBJECT_SCHEMA_READER_INITIALIZING

[wls_ods1] [NOTIFICATION] [DIP-10572] [oracle.dip] [tid: Scheduler]
[userId: <anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP] Object Schema
Reader Initialized.

[wls_ods1] [NOTIFICATION] [DIP-10573] [oracle.dip] [tid: Scheduler]
[userId: <anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP] Event Schema
Reader Initialized.

[wls_ods1] [NOTIFICATION] [DIP-10574] [oracle.dip] [tid: Scheduler]
[userId: <anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP] Data
transfer interface defn initialized

[wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid: Scheduler] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP] INITIALIZE_PROVJOBS

[wls_ods1] [NOTIFICATION] [DIP-10566] [oracle.dip] [tid: Scheduler]
[userId: <anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP] [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,orclprivilege,groupofuniquenames,groupofnames
-----
]]

[wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid: Scheduler] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP] INITIALIZE_SYNCJOBS

[wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid: Scheduler] [userId:
<anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP] Job submission
successfulActiveExport SYNC_JOB 60

[wls_ods1] [NOTIFICATION] [EVENT_NOT_ENABLED] [oracle.dip] [tid:
Scheduler] [userId: <anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP]

[wls_ods1] [NOTIFICATION] [DIP-10605] [oracle.dip] [tid: Scheduler]
[userId: <anonymous>] [ecid: 0000Hy8^kIXF0FQ6ubn3EH19awhV000001,0] [APP: DIP] [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:

```

[wls_ods1] [NOTIFICATION] [DIP-10580] [oracle.dip] [tid:
UpdateThread] [userId: <anonymous>] [ecid: 0000Hy8fyF1F0FQ6ubn3EH19ax8V000003,0] [APP:
DIP] [arg:
(&(objectclass=changelogentry)(changenumber>=3340)(
(targetdn=*cn=Profiles,cn=Provisioning,cn=Directory
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=Directory
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:

```

[wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid: Scheduler] [userId:
<anonymous>] [ecid: 0000Hy8unSqF0FQ6ubn3EH19ay88000001,0] [APP: DIP] INITIALIZE_SYNCJOBS

[wls_ods1] [NOTIFICATION] [] [oracle.dip] [tid: Scheduler] [userId:
<anonymous>] [ecid: 0000Hy8unSqF0FQ6ubn3EH19ay88000001,0] [APP: DIP] Job submission
successfulActiveImport SYNC_JOB 60

[wls_ods1] [NOTIFICATION] [EVENT_NOT_ENABLED] [oracle.dip] [tid:
Scheduler] [userId: <anonymous>] [ecid: 0000Hy8unSqF0FQ6ubn3EH19ay88000001,0] [APP: DIP]

[wls_ods1] [NOTIFICATION] [DIP-10605] [oracle.dip] [tid: Scheduler]

```

```
[userId: <anonymous>] [ecid: 0000Hy8unSqF0FQ6ubn3EH19ay88000001,0] [APP: DIP] [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:

```
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.
createChangeRecord:ChangeRecord : -----
Changetype: ADDRMODIFY
ChangeKey: cn=myuser2,cn=users,dc=intest,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.
  
```

## D.6 Troubleshooting Integration with Microsoft Active Directory

This section describes how to troubleshoot integration with Microsoft Active Directory.

### Topics:

- [Debugging Windows Native Authentication](#)
- [Synchronizing Changes Following a Period when the Oracle Back-end Directory is Unavailable](#)

### D.6.1 Debugging Windows Native Authentication

Know more on how to debug Windows Native Authentication.

Once you have configured Windows Native Authentication (see "[Configuring Access Manager for Windows Native Authentication](#)" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Access Management*), 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:

```
-Djazzn.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.example.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 -Djazzn.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://support.oracle.com/>.
- The "Problems and Solutions for Windows Native Authentication Errors" section in the Troubleshooting chapter of the *Oracle Fusion Middleware Enterprise Single Sign-On Suite Administrator's Guide* for more information about Windows Native Authentication errors.

## D.6.2 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]
Reader: oracle.ldap.odip.gsi.ActiveChgReader
SkipErrorToSyncNextChange: true
SearchDeltaSize: 100000
UpdateSearchCount: 100
```

 **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** tab section in “[Creating Synchronization Profiles](#)” 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. For example:

```
manageSyncProfiles update -h MYOIDHOST -p 7005 -D weblogic -pf
ActiveChgImp-ad-catchup -params "odip.profile.configfile
/home/oracle/product/OracleIDM/ldap/odi/conf/profile_config.txt"
```

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.

## D.7 Troubleshooting SSL/TLS

This section describes how to troubleshoot secure connection issues with either the Oracle back-end directory or connected directory.

Complete the following:

1. Create a boot identity file:
  - a. Open a text editor and name it as `boot.properties`.
  - b. Add the following code snippet:

```
username=<weblogic admin user name>
password=<weblogic admin user password>
```

- c. Save the file to the following location:
 

```
$DOMAIN_HOME/servers/wls_ods1/security/
```
2. Open the `startWebLogic.sh` script (Located at `$DOMAIN_HOME/bin/`) in a text editor.
3. Locate the following line:

```
JAVA_OPTIONS="{JAVA_OPTIONS}" -
Djava.util.logging.manager=oracle.core.ojdl.logging.ODLLogManager"
```

4. Replace it with the following line:

```
JAVA_OPTIONS="{JAVA_OPTIONS} -Djavax.net.debug=ssl -  
Djava.util.logging.manager=oracle.core.ojdl.logging.ODLLogManager"
```

5. Restart the Oracle Directory Integration Platform Managed Server (wls\_ods1):

```
$DOMAIN_HOME/bin/startManagedWeblogic.sh wls_ods1 | tee ssl.log
```

## D.8 Need More Help?

Access additional help through My Oracle Support and Oracle Directory Integration Platform Release Notes.

You can find more solutions in My Oracle Support (formerly MetaLink) at <http://support.oracle.com/>. If you do not find a solution for your problem, log a service request.

### See Also:

Oracle Directory Integration Platform in *Oracle Fusion Middleware Release Notes for Oracle Identity Management*.



# 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 references](#) (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 or Oracle Unified Directory. These applications are typically interested in changes to that information in Oracle Internet Directory or Oracle Unified 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.

# Index

## A

---

asynchronous provisioning, [15-5](#)

## D

---

Data Access Java plug-in, [15-4](#)

data flow, provisioning, [15-6](#)

## O

---

Oracle Directory Integration Platform, [15-2](#)

Oracle Identity Manager, [15-2](#)

## P

---

PL/SQL plug-in, [15-5](#)

plug-in

    Data Access Java, [15-4](#)

plug-in (*continued*)

    PL/SQL, [15-5](#)

    Pre-Data Entry, [15-6](#)

plug-in Post-Data Entry, [15-6](#)

Post-Data Entry plug-in, [15-6](#)

Pre-Data Entry plug-in, [15-6](#)

provisioning

    asynchronous, [15-5](#)

    provisioning integration profile, [15-3](#)

    synchronous, [15-4](#)

provisioning concepts, understanding, [15-3](#)

provisioning data flow, [15-6](#)

## S

---

synchronous provisioning, [15-4](#)

## U

---

understanding provisioning concepts, [15-3](#)