

Oracle® Identity Manager

Connector Guide for Database User Management



Release 11.1.1

E28315-15

April 2022

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

ORACLE®

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, Java, and MySQL 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	xx
Documentation Accessibility	xx
Related Documents	xx
Conventions	xx

What's New in the Oracle Identity Manager Connector for Database User Management?

Software Updates	xxii
Documentation-Specific Updates	xxvii

1 About the Connector

1.1	Introduction to the Database User Management Connector	1-1
1.2	Certified Components	1-2
1.3	Usage Recommendation	1-3
1.4	Certified Languages	1-3
1.5	Connector Architecture	1-4
1.6	Features of the Connector	1-7
1.6.1	Mapping Standard and Custom Attributes for Reconciliation and Provisioning	1-7
1.6.2	Predefined and Custom Reconciliation Queries	1-7
1.6.3	Predefined and Custom Provisioning Queries	1-8
1.6.4	Support for Creating Global and External Users In Oracle Database	1-8
1.6.5	Full and Incremental Reconciliation	1-9
1.6.6	Limited (Filtered) Reconciliation	1-9
1.6.7	Batched Reconciliation	1-9
1.6.8	Specifying Accounts to Be Excluded from Reconciliation and Provisioning Operations	1-10
1.6.9	Connection Pooling	1-10
1.6.10	Support for Connector Server	1-10
1.6.11	Support for Creating Connector Copies	1-11
1.6.12	Transformation and Validation of Account Data	1-11

1.6.13	Support for Reconciling Data About Deleted Entities	1-11
1.6.14	Separate Scheduled Jobs for Reconciliation of Users, Logins, and Deleted Login Entities	1-12
1.6.15	Support for SSL Communication Between the Target System and Oracle Identity Manager	1-12
1.6.16	Support for Managing Authorization to Oracle Database Vault Realms	1-12
1.6.17	Support for Configuring the Connector for Enterprise User Security	1-13
1.7	Roadmap for Deploying and Using the Connector	1-13

2 Deploying the Connector

2.1	Preinstallation	2-1
2.1.1	Files and Directories on the Installation Media	2-1
2.1.2	Preinstallation on Microsoft SQL Server	2-2
2.1.3	Creating a Target System User Account for Connector Operations	2-3
2.2	Installation	2-5
2.2.1	Understanding the Installation of the Connector	2-5
2.2.2	Installing the Connector in Oracle Identity Manager	2-6
2.2.3	Deploying the Connector Bundle in a Connector Server	2-8
2.2.3.1	Installing and Configuring the Connector Server	2-9
2.2.3.2	Running the Connector Server	2-10
2.2.3.3	Installing the Connector on the Connector Server	2-11
2.3	Postinstallation	2-12
2.3.1	Postinstallation on Oracle Identity Manager	2-12
2.3.1.1	Configuring the Target System As a Trusted Source	2-12
2.3.1.2	Changing to the Required Input Locale	2-13
2.3.1.3	Clearing Content Related to the Connector Resource Bundles from the Server Cache	2-13
2.3.1.4	Creating the Administrator Account on Oracle Database Vault	2-15
2.3.1.5	Setting up the Lookup Definition for Connection Pooling	2-15
2.3.1.6	Managing Logging for Oracle Identity Manager	2-16
2.3.1.7	Configuring Oracle Identity Manager Release 11.1.2 or Later	2-19
2.3.1.8	Localizing Field Labels in UI Forms for the Connector	2-22
2.3.2	Configuring the IT Resource for the Target System	2-24
2.3.3	Configuring the Connector to Support Multiple Versions of the Target System	2-26
2.3.4	Configuring the IT Resource for the Connector Server	2-28
2.3.5	Enabling Logging for Connector Server	2-30
2.4	Upgrading the Connector	2-31
2.4.1	Preupgrade Steps	2-31
2.4.2	Upgrade Steps	2-31
2.4.3	Postupgrade Steps	2-32
2.5	Postcloning Steps	2-35

2.5.1	Connector Objects	2-35
2.5.2	Postcloning Configuration for User Accounts	2-36

3 Using and Extending the Connector for MSSQL

3.1	Configuring Secure Communication Between MSSQL and Oracle Identity Manager	3-1
3.2	Guidelines on Determining Values for the JDBC URL and Connection Properties Parameters for MSSQL	3-3
3.3	Lookup Definitions for MSSQL	3-4
3.3.1	Lookup Definitions Synchronized with MSSQL	3-5
3.3.2	Lookup Definitions for Configurations for MSSQL	3-6
3.3.2.1	Lookup.DBUM.MSSQL.Configuration	3-7
3.3.2.2	Lookup.DBUM.MSSQL.Login.Configuration	3-7
3.3.2.3	Lookup.DBUM.MSSQL.UM.Configuration	3-8
3.3.2.4	Lookup.DBUM.MSSQL.Configuration.Trusted	3-8
3.3.2.5	Lookup.DBUM.MSSQL.UM.Configuration.Trusted	3-9
3.3.3	Lookup Definitions for Attribute Mappings for MSSQL Login Entity	3-9
3.3.3.1	Lookup.DBUM.MSSQL.Login.ProvAttrMap	3-9
3.3.3.2	Lookup.DBUM.MSSQL.Login.ReconAttrMap	3-10
3.3.3.3	Lookup.DBUM.MSSQL.Login.ReconDefaults	3-10
3.3.4	Lookup Definitions for Attribute Mappings for MSSQL User Entity	3-10
3.3.4.1	Lookup.DBUM.MSSQL.UM.ProvAttrMap	3-11
3.3.4.2	Lookup.DBUM.MSSQL.UM.ReconAttrMap	3-11
3.3.4.3	Lookup.DBUM.MSSQL.UM.ReconDefaults	3-11
3.3.4.4	Lookup.DBUM.MSSQL.UM.ReconDefaults.Trusted	3-12
3.3.4.5	Lookup.DBUM.MSSQL.UM.ReconAttrMap.Trusted	3-12
3.3.5	Lookup Definitions for Exclusion Lists for MSSQL	3-12
3.3.6	Lookup Definitions for Transformation of Data in MSSQL	3-13
3.3.7	Lookup Definition for Validation of Data in MSSQL	3-13
3.4	Scheduled Jobs for MSSQL	3-14
3.4.1	Scheduled Jobs for Lookup Field Synchronization for MSSQL	3-14
3.4.2	Attributes of the Scheduled Jobs for Reconciliation of MSSQL User and Login Entities	3-15
3.4.3	Attributes of the Scheduled Jobs for MSSQL Delete Operations	3-16
3.4.4	Configuring Scheduled Jobs for MSSQL	3-17
3.5	Reconciliation from MSSQL	3-18
3.5.1	Guidelines on Configuring Reconciliation from MSSQL	3-19
3.5.2	About the Reconciliation Process for MSSQL	3-19
3.5.3	About the Reconciliation Queries for MSSQL	3-20
3.5.4	Target System Columns Used in Reconciliation from MSSQL	3-21
3.5.5	Configuring the Target System As a Trusted Source	3-21
3.5.6	Reconciliation Rules for MSSQL	3-22

3.5.7	Viewing Reconciliation Rules for MSSQL	3-23
3.5.8	Reconciliation Action Rules for MSSQL	3-23
3.5.9	Viewing Reconciliation Action Rules for MSSQL	3-24
3.5.10	Performing Full Reconciliation from MSSQL	3-24
3.5.11	Performing Limited Reconciliation from MSSQL	3-25
3.5.12	Performing Incremental Reconciliation from MSSQL	3-25
3.6	Provisioning for MSSQL	3-25
3.6.1	Guidelines on Performing Provisioning Operations for MSSQL	3-25
3.6.2	Understanding the Provisioning Process for MSSQL	3-27
3.6.3	Configuring Direct Provisioning for MSSQL	3-28
3.6.4	Configuring Request-Based Provisioning for MSSQL	3-29
3.6.4.1	About Request Based Provisioning for MSSQL	3-29
3.6.4.2	Enabling Request-Based Provisioning	3-30
3.6.5	Switching Between Request-Based Provisioning and Direct Provisioning for MSSQL	3-32
3.6.5.1	Switching From Request-Based Provisioning to Direct Provisioning	3-32
3.6.5.2	Switching From Direct Provisioning to Request-Based Provisioning	3-33
3.6.6	Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x	3-33
3.7	Extending the Connector for MSSQL	3-35
3.7.1	Modifying Predefined Queries or Creating New Queries for MSSQL	3-35
3.7.1.1	About the Queries for MSSQL	3-35
3.7.1.2	Syntax of Provisioning Queries for MSSQL	3-36
3.7.1.3	Syntax of Reconciliation Queries for MSSQL	3-38
3.7.1.4	Syntax of List of Values Queries for MSSQL	3-40
3.7.1.5	Guidelines for Configuring Search Queries Used in Reconciliation from MSSQL	3-40
3.7.2	Configuring Queries to Add Support for Custom Parameters and Lookup Fields for MSSQL	3-41
3.7.2.1	Updating a Query File for MSSQL	3-41
3.7.2.2	Configuring Oracle Identity Manager	3-43
3.7.3	About Configuring the Connector for Multiple Installations of MSSQL	3-45
3.7.4	About Configuring the Connector for Multiple Trusted Source Reconciliation from MSSQL	3-48
3.7.5	Configuring Validation of Data During Reconciliation and Provisioning for MSSQL	3-49
3.7.6	Configuring Transformation of Data During User Reconciliation for MSSQL	3-51
3.7.7	Configuring Resource Exclusion Lists for MSSQL	3-53
3.7.8	Setting Up Action Scripts for MSSQL	3-54
3.7.8.1	About Action Scripts for MSSQL	3-54
3.7.8.2	Configuring Action Scripts for MSSQL	3-55

4 Using and Extending the Connector for Oracle Database

4.1	Configuring Secure Communication Between Oracle Database and Oracle Identity Manager	4-2
4.1.1	Configuring Data Encryption and Integrity in Oracle Database	4-2
4.1.2	Configuring SSL Communication in Oracle Database	4-2
4.2	Determining Values for the JDBC URL and Connection Properties Parameters for Oracle Database	4-3
4.2.1	Only Data Encryption and Integrity Is Configured	4-3
4.2.2	Only SSL Communication Is Configured	4-4
4.2.3	Both Data Encryption and Integrity and SSL Communication Are Configured	4-6
4.3	Lookup Definitions Used During Connector Operations for Oracle Database	4-7
4.3.1	Lookup Definitions Synchronized with Oracle Database	4-7
4.3.2	Lookup Definitions for Configurations for Oracle Database	4-9
4.3.2.1	Lookup.DBUM.Oracle.Configuration	4-9
4.3.2.2	Lookup.DBUM.Oracle.UM.Configuration	4-10
4.3.2.3	Lookup.DBUM.Oracle.Configuration.Trusted	4-10
4.3.2.4	Lookup.DBUM.Oracle.UM.Configuration.Trusted	4-11
4.3.3	Lookup Definitions for Attribute Mappings for Oracle Database	4-11
4.3.3.1	Lookup.DBUM.Oracle.UM.ProvAttrMap	4-11
4.3.3.2	Lookup.DBUM.Oracle.UM.ReconAttrMap	4-12
4.3.3.3	Lookup.DBUM.Oracle.UM.ReconAttrMap.Trusted	4-13
4.3.3.4	Lookup.DBUM.Oracle.UM.ReconDefaults.Trusted	4-13
4.3.4	Lookup Definitions for Exclusion Lists for Oracle Database	4-14
4.3.5	Lookup Definitions for Transformation of Data in Oracle Database	4-14
4.3.6	Lookup Definition for Validation of Data in Oracle Database	4-15
4.4	Scheduled Jobs for Oracle Database	4-15
4.4.1	Scheduled Jobs for Lookup Field Synchronization for Oracle Database	4-15
4.4.2	Attributes of the Scheduled Jobs for Oracle Database	4-16
4.4.3	Configuring Scheduled Jobs for Oracle Database	4-18
4.5	Reconciliation from Oracle Database	4-19
4.5.1	Guidelines on Configuring Reconciliation for Oracle Database	4-19
4.5.2	About Reconciliation Process for Oracle Database	4-20
4.5.3	About Reconciliation Queries for Oracle Database	4-21
4.5.4	Target System Columns Used in Reconciliation from Oracle Database	4-22
4.5.5	Configuring the Target System As a Trusted Source	4-23
4.5.6	Reconciliation Rules for Oracle Database	4-24
4.5.7	Viewing Reconciliation Rules for Oracle Database	4-24
4.5.8	Reconciliation Action Rules for Oracle Database	4-24
4.5.9	Viewing Reconciliation Action Rules for Oracle Database	4-25
4.5.10	Performing Full Reconciliation from Oracle Database	4-25
4.5.11	Performing Limited Reconciliation from Oracle Database	4-26

4.5.12	Performing Batched Reconciliation from Oracle Database	4-27
4.5.13	Performing Incremental Reconciliation from Oracle Database	4-27
4.6	Provisioning for Oracle Database	4-27
4.6.1	Guidelines on Performing Provisioning Operations for Oracle Database	4-28
4.6.2	Understanding the Provisioning Process for Oracle Database	4-29
4.6.3	Configuring Direct Provisioning for Oracle Database	4-31
4.6.4	Configuring Request-Based Provisioning for Oracle Database	4-32
4.6.4.1	About Request-Based Provisioning for Oracle Database	4-32
4.6.4.2	Enabling Request-Based Provisioning	4-33
4.6.5	Switching Between Request-Based Provisioning and Direct Provisioning for Oracle Database	4-35
4.6.5.1	Switching From Request-Based Provisioning to Direct Provisioning	4-35
4.6.5.2	Switching From Direct Provisioning to Request-Based Provisioning	4-36
4.6.6	Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x	4-36
4.7	Extending the Connector for Oracle Database	4-37
4.7.1	Modifying the Predefined Queries or Creating New Queries	4-38
4.7.1.1	About the Queries for Oracle Database	4-38
4.7.1.2	Syntax of Provisioning Queries for Oracle Database	4-39
4.7.1.3	Syntax of Reconciliation Queries for Oracle Database	4-41
4.7.1.4	Syntax of List of Values Queries for Oracle Database	4-43
4.7.1.5	Guidelines for Configuring Search Queries Used in Reconciliation from Oracle DatabaseS	4-43
4.7.2	Configuring Queries to Add Support for Custom Parameters and Lookup Fields for Oracle Database	4-44
4.7.2.1	Updating the Query Files for Oracle Database	4-44
4.7.2.2	Configuring Oracle Identity Manager	4-47
4.7.3	About Configuring the Connector for Multiple Installations of Oracle Database	4-48
4.7.4	About Configuring the Connector for Multiple Trusted Source Reconciliation from Oracle Database	4-52
4.7.5	Configuring Validation of Data During Reconciliation and Provisioning for Oracle Database	4-53
4.7.6	Configuring Transformation of Data During User Reconciliation for Oracle Database	4-55
4.7.7	Configuring Resource Exclusion Lists for Oracle Database	4-56
4.7.8	Setting Up Action Scripts for Oracle Database	4-58
4.7.8.1	About Action Scripts for Oracle Database	4-58
4.7.8.2	Configuring Action Scripts for Oracle Database	4-59

5 Using and Extending the Connector for MySQL

5.1	Configuring Secure Communication Between MySQL and Oracle Identity Manager	5-1
-----	--	-----

5.2	Guidelines on Determining Values for the JDBC URL and Connection Properties Parameters for MySQL	5-3
5.3	Lookup Definitions for MySQL	5-4
5.3.1	Lookup Definitions Synchronized with MySQL	5-4
5.3.2	Lookup Definitions for Configurations for MySQL	5-5
5.3.2.1	Lookup.DBUM.MySQL.Configuration	5-5
5.3.2.2	Lookup.DBUM.MySQL.UM.Configuration	5-6
5.3.2.3	Lookup.DBUM.MySQL.Configuration.Trusted	5-6
5.3.2.4	Lookup.DBUM.MySQL.UM.Configuration.Trusted	5-6
5.3.3	Lookup Definitions for Attribute Mappings for MySQL	5-7
5.3.3.1	Lookup.DBUM.MySQL.UM.ProvAttrMap	5-7
5.3.3.2	Lookup.DBUM.MySQL.UM.ReconAttrMap	5-7
5.3.3.3	Lookup.DBUM.MySQL.UM.ReconAttrMap.Trusted	5-8
5.3.3.4	Lookup.DBUM.MySQL.UM.ReconDefaults.Trusted	5-8
5.3.4	Lookup Definitions for Exclusion Lists for MySQL	5-8
5.3.5	Lookup Definitions for Transformation of Data in MySQL	5-9
5.3.6	Lookup Definition for Validation of Data in MySQL	5-9
5.4	Scheduled Jobs for MySQL	5-10
5.4.1	Scheduled Job for Lookup Field Synchronization for MySQL	5-10
5.4.2	Attributes for Scheduled Jobs for MySQL	5-11
5.4.3	Configuring Scheduled Jobs for MySQL	5-12
5.5	Reconciliation from MySQL	5-13
5.5.1	Guidelines on Configuring Reconciliation for MySQL	5-13
5.5.2	About the Reconciliation Process for MySQL	5-14
5.5.3	Target System Columns Used in Reconciliation from MySQL	5-15
5.5.4	Configuring the Target System As a Trusted Source	5-15
5.5.5	Reconciliation Rules for MySQL	5-16
5.5.6	Viewing Reconciliation Rules for MySQL	5-17
5.5.7	Reconciliation Action Rules for MySQL	5-17
5.5.8	Viewing Reconciliation Action Rules for MySQL	5-17
5.5.9	Performing Full Reconciliation from MySQL	5-18
5.5.10	Performing Limited Reconciliation from MySQL	5-18
5.5.11	Performing Batched Reconciliation from MySQL	5-19
5.6	Provisioning for MySQL	5-19
5.6.1	Guidelines on Performing Provisioning Operations for MySQL	5-19
5.6.2	Understanding the Provisioning Process for MySQL	5-20
5.6.3	Configuring Direct Provisioning for MySQL	5-21
5.6.4	Configuring Request-Based Provisioning for MySQL	5-22
5.6.4.1	About Request Based Provisioning for MySQL	5-22
5.6.4.2	Enabling Request-Based Provisioning	5-23
5.6.5	Switching Between Request-Based Provisioning and Direct Provisioning for MySQL	5-25

5.6.5.1	Switching From Request-Based Provisioning to Direct Provisioning	5-25
5.6.5.2	Switching From Direct Provisioning to Request-Based Provisioning	5-26
5.6.6	Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x	5-26
5.7	Extending the Connector for MySQL	5-27
5.7.1	Modifying Predefined Queries or Creating New Queries for MySQL	5-28
5.7.1.1	About the Queries for MySQL Database	5-28
5.7.1.2	Syntax of Provisioning Queries for MySQL Database	5-29
5.7.1.3	Syntax of Reconciliation Queries for MySQL Database	5-30
5.7.1.4	Syntax of List of Values Queries for MySQL Database	5-32
5.7.2	Configuring Queries to Add Support for Custom Parameters and Lookup Fields for MySQL	5-32
5.7.2.1	Updating the Query Files for MySQL Database	5-33
5.7.2.2	Configuring Oracle Identity Manager	5-35
5.7.3	About Configuring the Connector for Multiple Installations of MySQL	5-37
5.7.4	About Configuring the Connector for Multiple Trusted Source Reconciliation from MySQL	5-40
5.7.5	Configuring Validation of Data During Reconciliation and Provisioning for MySQL	5-41
5.7.6	Configuring Transformation of Data During User Reconciliation for MySQL	5-43
5.7.7	Configuring Resource Exclusion Lists for MySQL	5-44
5.7.8	Setting Up Action Scripts for MySQL	5-46
5.7.8.1	About Action Scripts for MySQL	5-46
5.7.8.2	Configuring Action Scripts for MySQL	5-47

6 Using and Extending the Connector for DB2

6.1	Configuring Secure Communication Between DB2 and Oracle Identity Manager	6-1
6.2	Guidelines on Determining Values for the JDBC URL and Connection Properties Parameters for DB2	6-2
6.3	Lookup Definitions for DB2	6-3
6.3.1	Lookup Definitions Synchronized with DB2	6-3
6.3.2	Lookup Definitions for Configurations for DB2	6-5
6.3.2.1	Lookup.DBUM.DB2.Configuration	6-5
6.3.2.2	Lookup.DBUM.DB2.UM.Configuration	6-6
6.3.2.3	Lookup.DBUM.DB2.Configuration.Trusted	6-6
6.3.2.4	Lookup.DBUM.DB2.UM.Configuration.Trusted	6-6
6.3.3	Lookup Definitions for Attribute Mappings for DB2	6-7
6.3.3.1	Lookup.DBUM.DB2.UM.ProvAttrMap	6-7
6.3.3.2	Lookup.DBUM.DB2.UM.ReconAttrMap	6-7
6.3.3.3	Lookup.DBUM.DB2.UM.ReconAttrMap.Trusted	6-8
6.3.3.4	Lookup.DBUM.DB2.UM.ReconDefaults.Trusted	6-8
6.3.4	Lookup Definitions for Exclusion Lists for DB2	6-9

6.3.5	Lookup Definitions for Transformation of Data in DB2	6-10
6.3.6	Lookup Definition for Validation of Data in DB2	6-10
6.4	Scheduled Jobs for DB2	6-10
6.4.1	Scheduled Jobs for Lookup Field Synchronization for DB2	6-10
6.4.2	Attributes for Scheduled Jobs for DB2	6-11
6.4.3	Configuring Scheduled Jobs for DB2	6-12
6.5	Reconciliation from DB2	6-13
6.5.1	Guidelines on Configuring Reconciliation for DB2	6-14
6.5.2	About the Reconciliation Process for DB2	6-14
6.5.3	Target System Columns Used in Reconciliation from DB2	6-15
6.5.4	Configuring the Target System As a Trusted Source	6-16
6.5.5	Reconciliation Rules for DB2	6-17
6.5.6	Viewing Reconciliation Rules for DB2	6-17
6.5.7	Reconciliation Action Rules for DB2	6-17
6.5.8	Viewing Reconciliation Action Rules for DB2	6-18
6.5.9	Performing Full Reconciliation from DB2	6-18
6.5.10	Performing Limited Reconciliation from DB2	6-19
6.5.11	Performing Batched Reconciliation from DB2	6-19
6.6	Provisioning for DB2	6-20
6.6.1	Guidelines on Performing Provisioning Operations for DB2	6-20
6.6.2	Understanding the Provisioning Process for DB2	6-20
6.6.3	Configuring Direct Provisioning for DB2	6-22
6.6.4	Configuring Request-Based Provisioning for DB2	6-23
6.6.4.1	About Request Based Provisioning for DB2	6-23
6.6.4.2	Enabling Request-Based Provisioning	6-24
6.6.5	Switching Between Request-Based Provisioning and Direct Provisioning for DB2	6-26
6.6.5.1	Switching From Request-Based Provisioning to Direct Provisioning	6-26
6.6.5.2	Switching From Direct Provisioning to Request-Based Provisioning	6-27
6.6.6	Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x	6-27
6.7	Extending the Connector for DB2	6-28
6.7.1	Modifying Predefined Queries or Creating New Queries for DB2	6-29
6.7.1.1	About the Queries for DB2	6-29
6.7.1.2	Syntax of Provisioning Queries for DB2	6-30
6.7.1.3	Syntax of Reconciliation Queries for DB2	6-32
6.7.1.4	Syntax of List of Values Queries for DB2	6-33
6.7.2	Configuring Queries to Add Support for Custom Parameters and Lookup Fields for DB2	6-34
6.7.2.1	Updating the Query Files for DB2	6-34
6.7.2.2	Configuring Oracle Identity Manager	6-36
6.7.3	About Configuring the Connector for Multiple Installations of DB2	6-38

6.7.4	About Configuring the Connector for Multiple Trusted Source Reconciliation from DB2	6-42
6.7.5	Configuring Validation of Data During Reconciliation and Provisioning for DB2	6-42
6.7.6	Configuring Transformation of Data During User Reconciliation for DB2	6-44
6.7.7	Configuring Resource Exclusion Lists for DB2	6-46
6.7.8	Setting Up Action Scripts for DB2	6-47
6.7.8.1	About Action Scripts for DB2	6-48
6.7.8.2	Configuring Action Scripts for DB2	6-48

7 Using and Extending the Connector for Sybase

7.1	Configuring Secure Communication Between Sybase and Oracle Identity Manager	7-1
7.2	Guidelines on Determining Values for the JDBC URL and Connection Properties Parameters for Sybase	7-2
7.3	Lookup Definitions for Sybase	7-3
7.3.1	Lookup Definitions Synchronized with Sybase	7-3
7.3.2	Lookup Definitions for Configurations for Sybase	7-4
7.3.2.1	Lookup.DBUM.Sybase.Configuration	7-5
7.3.2.2	Lookup.DBUM.Sybase.Login.Configuration	7-5
7.3.2.3	Lookup.DBUM.Sybase.UM.Configuration	7-6
7.3.2.4	Lookup.DBUM.Sybase.Configuration.Trusted	7-6
7.3.2.5	Lookup.DBUM.Sybase.UM.Configuration.Trusted	7-7
7.3.3	Lookup Definitions for Attribute Mappings for Sybase Login Entity	7-7
7.3.3.1	Lookup.DBUM.Sybase.Login.ProvAttrMap	7-7
7.3.3.2	Lookup.DBUM.Sybase.Login.ReconAttrMap	7-8
7.3.3.3	Lookup.DBUM.Sybase.Login.ReconDefaults	7-8
7.3.4	Lookup Definitions for Attribute Mappings for Sybase User Entity	7-8
7.3.4.1	Lookup.DBUM.Sybase.UM.ProvAttrMap	7-8
7.3.4.2	Lookup.DBUM.Sybase.UM.ReconAttrMap	7-9
7.3.4.3	Lookup.DBUM.Sybase.UM.ReconDefaults	7-9
7.3.4.4	Lookup.DBUM.Sybase.UM.ReconDefaults.Trusted	7-9
7.3.4.5	Lookup.DBUM.Sybase.UM.ReconAttrMap.Trusted	7-10
7.3.5	Lookup Definition for Validation of Data in Sybase	7-10
7.3.6	Lookup Definitions for Transformation of Data in Sybase	7-10
7.3.7	Lookup Definitions for Exclusion Lists for Sybase	7-11
7.4	Scheduled Jobs for Sybase	7-11
7.4.1	Scheduled Jobs for Lookup Field Synchronization for Sybase	7-11
7.4.2	Attributes for Scheduled Jobs for Sybase	7-12
7.4.3	Configuring Scheduled Jobs for Sybase	7-14
7.5	Reconciliation from Sybase	7-15
7.5.1	About the Reconciliation Process for Sybase	7-15
7.5.2	Reconciliation Rules for Sybase	7-16

7.5.3	Viewing Reconciliation Rules for Sybase	7-17
7.5.4	Reconciliation Action Rules for Sybase	7-17
7.5.5	Viewing Reconciliation Action Rules for Sybase	7-17
7.5.6	Performing Full Reconciliation from Sybase	7-18
7.6	Provisioning for Sybase	7-18
7.6.1	Guidelines on Performing Provisioning Operations for Sybase	7-18
7.6.2	Understanding the Provisioning Process for Sybase	7-19
7.6.3	Configuring Direct Provisioning for Sybase	7-21
7.6.4	Configuring Request-Based Provisioning for Sybase	7-21
7.6.4.1	About Request Based Provisioning for Sybase	7-22
7.6.4.2	Enabling Request-Based Provisioning	7-22
7.6.5	Switching Between Request-Based Provisioning and Direct Provisioning for Sybase	7-24
7.6.5.1	Switching From Request-Based Provisioning to Direct Provisioning	7-25
7.6.5.2	Switching From Direct Provisioning to Request-Based Provisioning	7-25
7.6.6	Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x	7-26
7.7	Extending the Connector for Sybase	7-27
7.7.1	Modifying Predefined Queries or Creating New Queries for Sybase	7-27
7.7.1.1	About the Queries for Sybase	7-27
7.7.1.2	Syntax of Provisioning Queries for Sybase Database	7-28
7.7.1.3	Syntax of Reconciliation Queries for Sybase Database	7-30
7.7.1.4	Syntax of List of Values Queries for Sybase Database	7-32
7.7.2	Configuring Queries to Add Support for Custom Parameters and Lookup Fields for Sybase	7-32
7.7.2.1	Updating the Query Files for Sybase Database	7-32
7.7.2.2	Configuring Oracle Identity Manager	7-35
7.7.3	About Configuring the Connector for Multiple Installations of Sybase	7-36
7.7.4	About Configuring the Connector for Multiple Trusted Source Reconciliation from Sybase	7-39
7.7.5	Configuring Validation of Data During Reconciliation and Provisioning for Sybase	7-40
7.7.6	Configuring Transformation of Data During User Reconciliation for Sybase	7-42
7.7.7	Configuring Resource Exclusion Lists for Sybase	7-44
7.7.8	Setting Up Action Scripts for Sybase	7-45
7.7.8.1	About Action Scripts for Sybase	7-46
7.7.8.2	Configuring Action Scripts for Sybase	7-46

8 Configuring the Connector for a JDBC-Based Database

8.1	Target System Attributes and Queries	8-1
8.2	Configuring the Queries	8-2
8.3	Updating the Query Files to the Connector Bundle	8-4

8.4	Configuring the IT Resource	8-6
8.5	Configuring the Process Form	8-7
8.6	Configuring the Resource Object	8-9
8.7	Adding Process Tasks, Assigning Adapters, and Mapping Adapter Variables	8-10
8.8	Adding Attributes for Reconciliation	8-13
8.9	Configuring Lookup Definitions Used During Connector Operations	8-14
8.10	Configuring Scheduled Jobs	8-16
8.10.1	Configuring Scheduled Jobs for Lookup Field Synchronization	8-16
8.10.2	Configuring Scheduled Jobs for Reconciliation	8-17
8.11	Configuring Oracle Identity Manager for Request-Based Provisioning	8-18
8.12	Testing the Customized Connector	8-19

9 Testing the Connector

9.1	Running the Test Utility for the Oracle Database	9-1
9.2	Running the Test Utility for the MSSQL Database	9-3
9.3	Running the Test Utility for MySQL	9-5
9.4	Running the Test Utility for DB2	9-6
9.5	Running the Test Utility for Sybase	9-8

10 Known Issues and Workarounds

10.1	Connector Issues	10-1
10.1.1	Wild Card Expressions not Supported in Privileges for MySQL	10-1
10.2	Oracle Identity Manager Issues	10-1
10.2.1	Update of a Child Table does not Work as Expected	10-1
10.2.2	User Remains in Provisioned Status After a Trusted Delete Reconciliation Run	10-2
10.2.3	Target Reconciliation Does Not Revoke Roles and Privileges	10-2
10.2.4	The With Grant Option in Child Data Fails to Appear in Entitlement List	10-2
10.2.5	The Second Installation of the Connector Fails	10-2
10.3	Target System Issues	10-2
10.3.1	Creation of a JDBC Connection Results in an Indefinite Wait Time	10-3
10.3.2	Incremental Reconciliation Cannot Process Role Updates in MSSQL	10-3
10.3.3	Relevant Error Log not Provided While Provisioning a Duplicate User	10-3
10.3.4	Provisioning of Privileges for the Schema not Supported for MySQL	10-3

A Configuring Requests

A.1	Step 1: Creating a Request Dataset for the Resources	A-1
A.1.1	About Request Datasets	A-1
A.1.2	Default Request Datasets	A-2

A.1.3	Elements and Properties	A-3
A.1.3.1	The request-data-set Element	A-4
A.1.3.2	The DataSetValidator Element	A-5
A.1.3.3	The AttributeReference Element	A-5
A.1.3.4	The Attribute Element	A-12
A.1.4	Sample Request Dataset	A-13
A.1.5	Child Data	A-16
A.1.6	Common Request Dataset	A-20
A.1.7	Configuring Localized Values for Request Datasets	A-21
A.1.7.1	Localization for Request Dataset Attributes	A-21
A.1.7.2	Localization of Column Names in LookupQuery for Dataset Attributes	A-24
A.2	Step 2: Uploading Request Datasets into MDS	A-24
A.3	Step 3: Creating SOA Composites Required for Approval	A-27
A.3.1	Using SOA Composites for Approval	A-27
A.3.2	Creating a SOA Composite	A-28
A.4	Step 4: Registering the SOA Composites in Oracle Identity Manager	A-30
A.5	Step 5: Defining Request Approvals	A-31
A.5.1	Approval Workflows	A-32
A.5.2	Approval Levels	A-33
A.5.2.1	Template-Level Approval	A-33
A.5.2.2	Request-Level Approval	A-34
A.5.2.3	Operation-Level Approval	A-35
A.5.3	Creating Approval Policies	A-36
A.6	Step 6: Creating Request Templates	A-38
A.6.1	About Request Templates	A-38
A.6.2	Creating a Request Template Based on the Create User Request Type	A-40
A.6.3	Creating a Request Template Based on the Provisioning Resource Request Type	A-47
A.7	Extending Request Management Operations	A-50
A.7.1	Running Custom Code Based on Request Status Change	A-50
A.7.2	Validating Request Data	A-52
A.7.3	Prepopulation of an Attribute Value During Request Creation	A-52

Index

Index

List of Figures

1-1	Architecture of the Connector	1-4
2-1	Manage IT Resource Page for Connector Server IT Resource	2-29
2-2	Edit IT Resource Details and Parameters Page for the Connector Server IT Resource	2-29
A-1	Request Service and SOA Integration	A-32
A-2	The Set Request Template Details Page	A-41
A-3	The Select Attributes to Restrict Page	A-42
A-4	The Set Attribute Restrictions Page	A-44
A-5	The Set Additional Attributes Page	A-45
A-6	The Set Template User Roles Page	A-46
A-7	The Review Request Template Summary Page	A-47

List of Tables

1-1	Certified Components	1-2
2-1	Files and Directories on the Installation Media	2-1
2-2	Connection Pooling Properties	2-15
2-3	Log Levels and ODL Message Type:Level Combinations	2-17
2-4	IT Resource Parameters	2-25
2-5	Parameters of the IT Resource for the DBUM Connector Server	2-30
2-6	Mappings for DBUM Oracle Event Handler/Adapter	2-37
3-1	Truststore Locations on Supported Application Servers	3-2
3-2	Lookup Definitions Synchronized with Microsoft SQL Server	3-6
3-3	Entries in the Lookup.DBUM.MSSQL.AuthType Lookup Definition	3-6
3-4	Entries in the Lookup.DBUM.MSSQL.Configuration Lookup Definition	3-7
3-5	Entries in Lookup.DBUM.MSSQL.Login.Configuration	3-7
3-6	Entries in Lookup.DBUM.MSSQL.UM.Configuration	3-8
3-7	Entries in the Lookup.DBUM.MSSQL.Configuration.Trusted Lookup Definition	3-9
3-8	Entries in Lookup.DBUM.MSSQL.UM.Configuration.Trusted	3-9
3-9	Entries in Lookup.DBUM.MSSQL.Login.ProvAttrMap	3-10
3-10	Entries in Lookup.DBUM.MSSQL.Login.ReconAttrMap	3-10
3-11	Entries in Lookup.DBUM.MSSQL.UM.ProvAttrMap	3-11
3-12	Entries in Lookup.DBUM.MSSQL.UM.ReconAttrMap	3-11
3-13	Entries in Lookup.DBUM.MSSQL.UM.ReconDefaults	3-11
3-14	Entries in Lookup.DBUM.MSSQL.UM.ReconDefaults.Trusted	3-12
3-15	Entries in Lookup.DBUM.MSSQL.UM.ReconAttrMap.Trusted	3-12
3-16	Entries in Lookup.DBUM.MSSQL.Login.ReconTransformation	3-13
3-17	Attributes of the Scheduled Jobs for Lookup Field Synchronization for MSSQL	3-14
3-18	Attributes of the Scheduled Jobs for Reconciliation for MSSQL User and Login Entities	3-15
3-19	Attributes of the Scheduled Jobs for Delete Operations	3-17
3-20	Action Rules for Target Resource Reconciliation for MSSQL	3-23
3-21	Action Rules for Trusted Source Reconciliation for MSSQL	3-24
3-22	Script Section Logic for MSSQL Provisioning Queries	3-38
3-23	Connector Objects and Their Associations	3-46
4-1	Truststore Locations on Supported Application Servers	4-3
4-2	Lookup Definitions Synchronized with Oracle Database	4-8
4-3	Entries in Lookup.DBUM.Oracle.AuthType	4-9
4-4	Entries in Lookup.DBUM.Oracle.Configuration	4-9
4-5	Entries in Lookup.DBUM.Oracle.UM.Configuration	4-10

4-6	Entries in Lookup.DBUM.Oracle.Configuration.Trusted	4-11
4-7	Entries in Lookup.DBUM.Oracle.UM.Configuration.Trusted	4-11
4-8	Entries in Lookup.DBUM.Oracle.UM.ProvAttrMap	4-11
4-9	Entries in Lookup.DBUM.Oracle.UM.ReconAttrMap	4-12
4-10	Entries in Lookup.DBUM.Oracle.UM.ReconAttrMap.Trusted	4-13
4-11	Entries in Lookup.DBUM.Oracle.UM.ReconDefaults.Trusted	4-13
4-12	Entries in Lookup.DBUM.Oracle.UM.ReconTransformations	4-15
4-13	Attributes of the Scheduled Jobs for Lookup Field Synchronization	4-16
4-14	Attributes of the Scheduled Jobs for Reconciliation	4-17
4-15	Attributes of the Scheduled Jobs for Delete Operations	4-17
4-16	Action Rules for Target Resource Reconciliation	4-25
4-17	Action Rules for Trusted Source Reconciliation	4-25
4-18	Script Section Logic for Oracle Provisioning Queries	4-40
4-19	Connector Objects and Their Associations	4-50
5-1	Truststore Locations on Supported Application Servers	5-2
5-2	Lookup Definitions Synchronized with MySQL	5-5
5-3	Entries in Lookup.DBUM.MySQL.Configuration	5-5
5-4	Entries in Lookup.DBUM.MySQL.UM.Configuration	5-6
5-5	Entries in Lookup.DBUM.MySQL.Configuration.Trusted	5-6
5-6	Entries in Lookup.DBUM.MySQL.UM.Configuration.Trusted	5-7
5-7	Entries in Lookup.DBUM.MySQL.UM.ProvAttrMap	5-7
5-8	Entries in Lookup.DBUM.MySQL.UM.ReconAttrMap	5-7
5-9	Entries in Lookup.DBUM.MySQL.UM.ReconAttrMap.Trusted	5-8
5-10	Entries in Lookup.DBUM.MySQL.UM.ReconDefaults.Trusted	5-8
5-11	Attributes of the Scheduled Job for Lookup Field Synchronization	5-10
5-12	Attributes of the Scheduled Jobs for Reconciliation	5-11
5-13	Attributes of the Scheduled Jobs for Delete Operations	5-12
5-14	Action Rules for Target Resource Reconciliation	5-17
5-15	Action Rules for Trusted Source Reconciliation	5-17
5-16	Script Section Logic for MySQL Provisioning Queries	5-30
5-17	Connector Objects and Their Associations	5-38
6-1	Truststore Locations on Supported Application Servers	6-2
6-2	Lookup Definitions Synchronized with DB2	6-4
6-3	Entries in Lookup.DBUM.DB2.Configuration	6-5
6-4	Entries in Lookup.DBUM.DB2.UM.Configuration	6-6
6-5	Entries in Lookup.DBUM.DB2.Configuration.Trusted	6-6
6-6	Entries in Lookup.DBUM.DB2.UM.Configuration.Trusted	6-6

6-7	Entries in Lookup.DBUM.DB2.UM.ProvAttrMap	6-7
6-8	Entries in Lookup.DBUM.DB2.UM.ReconAttrMap	6-8
6-9	Entries in Lookup.DBUM.DB2.UM.ReconAttrMap.Trusted	6-8
6-10	Entries in Lookup.DBUM.DB2.UM.ReconDefaults.Trusted	6-8
6-11	Attributes of the Scheduled Jobs for Lookup Field Synchronization	6-11
6-12	Attributes of the Scheduled Jobs for Reconciliation	6-12
6-13	Action Rules for Target Resource Reconciliation	6-18
6-14	Action Rules for Trusted Source Reconciliation	6-18
6-15	Script Section Logic for DB2 Provisioning Queries	6-31
6-16	Connector Objects and Their Associations	6-39
7-1	Truststore Locations on Supported Application Servers	7-2
7-2	Lookup Definitions Synchronized with Sybase	7-4
7-3	Entries in Lookup.DBUM.Sybase.Configuration	7-5
7-4	Entries in Lookup.DBUM.Sybase.Login.Configuration	7-5
7-5	Entries in Lookup.DBUM.Sybase.UM.Configuration	7-6
7-6	Entries in the Lookup.DBUM.Sybase.Configuration.Trusted Lookup Definition	7-6
7-7	Entries in Lookup.DBUM.Sybase.UM.Configuration.Trusted	7-7
7-8	Entries in Lookup.DBUM.Sybase.Login.ProvAttrMap	7-7
7-9	Entries in Lookup.DBUM.Sybase.Login.ReconAttrMap	7-8
7-10	Entries in Lookup.DBUM.Sybase.Login.ReconDefaults	7-8
7-11	Entries in Lookup.DBUM.Sybase.UM.ProvAttrMap	7-9
7-12	Entries in Lookup.DBUM.Sybase.UM.ReconAttrMap	7-9
7-13	Entries in Lookup.DBUM.Sybase.UM.ReconDefaults	7-9
7-14	Entries in Lookup.DBUM.Sybase.UM.ReconDefaults.Trusted	7-10
7-15	Entries in Lookup.DBUM.Sybase.UM.ReconAttrMap.Trusted	7-10
7-16	Entries in Lookup.DBUM.Sybase.Login.ReconTransformations	7-10
7-17	Attributes of the Scheduled Jobs for Lookup Field Synchronization	7-12
7-18	Attributes of the Scheduled Jobs for Reconciliation	7-13
7-19	Attributes of the Scheduled Jobs for Delete Operations	7-13
7-20	Action Rules for Target Resource Reconciliation	7-17
7-21	Script Section Logic for Sybase Provisioning Queries	7-30
7-22	Connector Objects and Their Associations	7-37
8-1	IT Resource Parameters for New Database	8-6
A-1	Default Request Datasets Shipped with Oracle Identity Manager	A-2
A-2	Request Datasets for Resource Entity	A-4
A-3	Parameters in the Properties File	A-25
A-4	Request Types and Associated keys	A-35

Preface

This guide describes the connector that is used to set up Oracle Identity Manager for database user management.

Audience

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

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

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

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

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

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

Conventions

The following text conventions are used in this document:

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

Convention	Meaning
<code>monospace</code>	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 the Oracle Identity Manager Connector for Database User Management?

This chapter provides an overview of the updates made to the software and documentation for release 11.1.1.8.0 of the Database User Management connector.



Note:

Release 11.1.1.8.0 of the connector comes after release 11.1.1.6.0. Release number 11.1.1.7.0 has not been used.

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

- [Software Updates](#)
These include updates made to the connector software.
- [Documentation-Specific Updates](#)
These include major changes made to the connector documentation. These changes are not related to software updates.

Software Updates

The following sections discuss software updates:

- [Software Updates in Release 11.1.1.8.0](#)
- [Software Updates in Release 11.1.1.6.0](#)
- [Software Updates in Release 11.1.1.5.0](#)

Software Updates in Release 11.1.1.8.0

The following are issues resolved in release 11.1.1.8.0:

Bug Number	Issue	Resolution
16476930 16476913 16476900 16476884 16474781	<p>The Entitlement, AccountName, and AccountID properties of fields on the process form for all target systems supported by this connector were not tagged.</p> <p>The "Entitlement" property of child form attributes was not set to <code>true</code>. This resulted in attributes not being displayed in the catalog. This prevented users from adding such entitlements provided by the connector to the shopping cart.</p> <p>The "AccountName" property of a process form field that represented the login ID of an account in the target system was not set to <code>true</code>. As a result of this, the Account Name column on the Accounts tab of the My Access page in the Self Service console for a user displayed the database numeric key, instead of the correct account name.</p> <p>The "AccountID" property of a process form field that represented the immutable GUID of the account (if one exists) was not set to <code>true</code>. This prevented integrating Oracle Identity Manager with Oracle Identity Analytics.</p>	<p>This issue has been resolved. The Entitlement, AccountName, and AccountID properties of fields on the process form have been tagged. In other words, the values of the Entitlement, AccountName, and AccountID properties have been set to <code>true</code>.</p>
16491431	<p>The following issue was observed if you are using MySQL as the target system:</p> <p>The Revoke Privilege provisioning task had a dependency on the Add Privilege provisioning task. As a result, if you granted multiple privileges to a user and one of the Add Privilege tasks did not complete successfully, then none of the assigned privileges were revoked. In addition, the Revoke Privilege task remained in the "Waiting" state.</p>	<p>This issue has been resolved.</p>
15955807	<p>The connector failed to update password in UpdateApiOp with MySQL database. In addition, no errors were reported.</p>	<p>This issue has been resolved. The connector updates the password in UpdateApiOp with MySQL database.</p>
16085209	<p>The following exception was encountered when you ran the connector test:</p> <p>DBUM Connector Test Fails JRF Portability Layer exception</p>	<p>This issue has been resolved.</p>

Bug Number	Issue	Resolution
15919151	The connector failed in SearchApiOp with MySQL database running on UNIX computer. This failure was due to table names being case sensitive in UNIX MySQL.	This issue has been resolved.
15895008	Changing the password of a user account in Sybase resulted in the failure of a stored procedure.	This issue has been resolved.

Software Updates in Release 11.1.1.6.0

The following is a software update in release 11.1.1.6.0:

Support for MySQL, DB2, and Sybase Databases

From this release onward, the connector supports the following additional databases:

- MySQL
See [Using and Extending the Connector for MySQL](#) for more information.
- DB2
See [Using and Extending the Connector for DB2](#) for more information.
- Sybase
See [Using and Extending the Connector for Sybase](#) for more information.

Software Updates in Release 11.1.1.5.0

The following are the software updates in release 11.1.1.5.0:

- [ICF Based Connector](#)
- [Connector Bundle](#)
- [Support for Connection Pooling](#)
- [Support for Running the Connector Remotely Using ICF Remote Connector Server](#)
- [Support for Multiple Instances and Multiple Versions of Target System](#)
- [Support for Running Pre/Post Action Scripts](#)
- [Support for Transformation and Validation](#)
- [Support for Resource Exclusion Lists](#)
- [Improved Trusted Reconciliation](#)
- [Support for Incremental and Limited Reconciliation](#)
- [Support for JDBC-Based Database Customization](#)
- [Support for MySQL, DB2, and Sybase Databases](#)

ICF Based Connector

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

The Oracle Identity Manager Connector for Database User Management is an ICF-based connector. The ICF uses classpath isolation, which allows the Database User Management connector to co-exist with legacy versions of the connector.

For more information about the ICF, see *Understanding the Identity Connector Framework in Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

Connector Bundle

From this release onward, all connector operations such as provisioning, reconciliation, and search, are performed by running SQL scripts stored in connector bundle.

In the previous release, reconciliation scripts were present in the filesystem and changes were expected for converting from full to incremental reconciliation.

From this release onwards, the script execution is more intelligent. It dynamically switches to incremental mode after first execution. In addition, no changes are required on script for limited reconciliation. They can all be managed from scheduled jobs.

Removing script loading from filesystem also reduces file synchronization issues in cluster deployment scenarios.

See [Connector Architecture](#) for more information about the connector bundle.

Support for Connection Pooling

From this release onward, connection pooling is supported. Oracle Identity Manager connectors can use these connections to communicate with target systems. See [Connection Pooling](#) for more information about connection pooling.

Support for Running the Connector Remotely Using ICF Remote Connector Server

This release of the connector supports running the connector remotely using ICF remote connector server. See [Installing the Connector on the Connector Server](#) in the connector guide for more information.

Support for Multiple Instances and Multiple Versions of Target System

This release of the connector supports multiple instances and multiple versions of target system. You can deploy a single connector bundle on Oracle Identity Manager and create multiple IT resources for multiple instances and multiple versions of a target system. Then, you can use Oracle Identity Manager to manage accounts on these target systems.

See the following sections in the connector guide for more information:

- [Configuring the Connector to Support Multiple Versions of the Target System](#)
- [About Configuring the Connector for Multiple Installations of Oracle Database](#)

- [About Configuring the Connector for Multiple Installations of MSSQL](#)

Support for Running Pre/Post Action Scripts

From this release onward, you can run pre/post action scripts on a computer where the DBUM connector is deployed. These scripts can be of type SQL/StoredProc/Groovy. You can configure the scripts to run before or after the create, update, or delete an account provisioning operations. See the following sections for more information:

- [Provisioning for Oracle Database](#)
- [Provisioning for MSSQL](#)

Support for Transformation and Validation

You can configure transformation of data, such as process form field data or any other object, that is brought into Oracle Identity Manager during reconciliation. In addition, you can configure validation of data that is brought into or sent from Oracle Identity Manager during reconciliation and provisioning. See the following sections for more information:

- [Configuring Transformation of Data During User Reconciliation for Oracle Database](#)
- [Configuring Validation of Data During Reconciliation and Provisioning for Oracle Database](#)
- [Configuring Transformation of Data During User Reconciliation for MSSQL](#)
- [Configuring Validation of Data During Reconciliation and Provisioning for MSSQL](#)

Support for Resource Exclusion Lists

From this release onward, you can specify a list of accounts that must be excluded from reconciliation and provisioning operations. Accounts whose user IDs you specify in the exclusion list are not affected by reconciliation and provisioning operations.

See the following sections in the connector guide for more information:

- [Configuring Resource Exclusion Lists for Oracle Database](#)
- [Configuring Resource Exclusion Lists for MSSQL](#)

Improved Trusted Reconciliation

From this release onward, the connector artifacts are provided OOB for trusted reconciliation. The incremental reconciliation is more simplified as user does not need to change queries file anymore to switch to incremental mode.

Support for Incremental and Limited Reconciliation

The release 11.1.1.5.0 supports incremental and limited reconciliation for the MSSQL database. See [Reconciliation from MSSQL](#) for more information.

Support for JDBC-Based Database Customization

The connector supports JDBC Based Database Customization against your target system. You can configure the connector to support the JDBC-based database by performing the required instructions. See [Configuring the Connector for a JDBC-Based Database](#) for more information about required instructions.

Documentation-Specific Updates

The following sections discuss documentation-specific updates:

- [Documentation-Specific Updates in Release 11.1.1.8.0](#)
- [Documentation-Specific Updates in Release 11.1.1.6.0](#)
- [Documentation-Specific Updates in Release 11.1.1.5.0](#)

Documentation-Specific Updates in Release 11.1.1.8.0

The following is a documentation-specific update in revision "14" of release 11.1.1.8.0:

The description for reservedWordsList and unsupportedChars Code Keys have been modified in [Table 4-4](#).

The following is a documentation-specific update in revision "13" of release 11.1.1.8.0:

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

The following is a documentation-specific update in revision "12" of release 11.1.1.8.0:

Information specific to Oracle Identity Manager 11g Release 2 PS3 (11.1.2.3.0) has been modified in [Usage Recommendation](#) to include the phrase "any later BP in this release track".

The following are documentation-specific update in revision "11" of release 11.1.1.8.0:

- The "Target systems" row of [Table 1-1](#) has been modified to include support for Microsoft SQL server 2016.
- [Creating a Target System User Account for Connector Operations](#) has been added.
- All the cross-references to the related Oracle Identity Manager guides have been updated with cross-references to the specific sections of those guides.
- Appendix A, "Configuring Requests for Database User Management Connector" has been added.
- The "[Related Documents](#)" section in [Preface](#) has been updated.

The following are documentation-specific updates in revision "10" of release 11.1.1.8.0:

- The "Connector Server" row has been added to [Table 1-1](#).
- The "JDK" row of [Table 1-1](#) has been renamed to "Connector Server JDK".
- [Postcloning Configuration for User Accounts](#) has been added.

The following are documentation-specific updates in revision "9" of release 11.1.1.8.0:

- The "Oracle Identity Manager" row of [Table 1-1](#) has been updated.

- The "Target systems" row of [Table 1-1](#) has been updated.
- Information specific to Oracle Identity Manager 11g Release 2 PS3 (11.1.2.3.0) has been added to [Usage Recommendation](#).
- The "Groovy" sample value has been removed from Step 3 of the following sections:
 - [Setting Up Action Scripts for MSSQL](#)
 - [Configuring Action Scripts for Oracle Database](#)
 - [Setting Up Action Scripts for MySQL](#)
 - [Setting Up Action Scripts for DB2](#)
 - [Setting Up Action Scripts for Sybase](#)
- The following changes have been made for the addition of information specific to Database ID:
 - The "Call UPDATE_DBID(usrid, dbid)" procedure has been added to the list of stored procedures that are used in the provisioning queries in [Target System Attributes and Queries](#).
 - Information specific to "UPDATE_DBID" query has been added to the information in the "Provisioning.queries" file in [Configuring the Queries](#).
 - The sample screenshot of the updated process task has been modified in [Adding Process Tasks, Assigning Adapters, and Mapping Adapter Variables](#).
 - Information specific to the update task for the Database ID field has been added at the end of [Adding Attributes for Reconciliation](#).
- A "Note" with information specific to lookup queries has been added at the beginning of the following sections:
 - [Extending the Connector for MSSQL](#)
 - [Extending the Connector for Oracle Database](#)
 - [Extending the Connector for MySQL](#)
 - [Extending the Connector for DB2](#)
 - [Extending the Connector for Sybase](#)

The following are documentation-specific updates in revision "8" of release 11.1.1.8.0:

- Oracle Database 12c and Microsoft SQL Server 2012 have been added to the "Target Systems" row of [Table 1-1](#).
- Information about Oracle 12c has been added to the "JDBC URL" row of [Table 2-4](#).
- Information about Microsoft SQL Server 2012 has been added to [Configuring the Connector to Support Multiple Versions of the Target System](#).
- Information about limited reconciliation has been modified in the following sections:
 - [Performing Limited Reconciliation from MSSQL](#)
 - [Performing Limited Reconciliation from Oracle Database](#)
 - [Performing Limited Reconciliation from MySQL](#)
 - [Performing Limited Reconciliation from DB2](#)

The following are documentation-specific updates in revision "7" of release 11.1.1.8.0:

- Information about making changes to the form designer was added to [Postupgrade Steps](#).
- Information was updated under bug number 16713445 in [Known Issues and Workarounds](#).
- Information about exclusion lists has been modified in the following sections:
 - [Configuring Resource Exclusion Lists for Oracle Database](#)
 - [Configuring Resource Exclusion Lists for MSSQL](#)
 - [Configuring Resource Exclusion Lists for MySQL](#)
 - [Configuring Resource Exclusion Lists for DB2](#)
 - [Configuring Resource Exclusion Lists for Sybase](#)
- Information about including the jrf.jar, jrf-api.jar, and jrf-client.jar files for Oracle Identity Manager release 11.1.2.x has been added as step 2 in the following sections:
 - [Running the Test Utility for the Oracle Database](#)
 - [Running the Test Utility for the MSSQL Database](#)
 - [Running the Test Utility for MySQL](#)
 - [Running the Test Utility for DB2](#)
 - [Running the Test Utility for Sybase](#)
- The name of the "Known Issues" chapter has been changed to "Known Issues and Workarounds." In addition, [Known Issues and Workarounds](#) has been restructured.

The following are documentation-specific updates in revision "6" of release 11.1.1.8.0:

- The "Oracle Identity Manager" row in [Table 1-1](#) has been modified.
- A note has been added in the "Files in the xml directory" row of [Table 2-1](#).
- The following sections have been added:
 - [Usage Recommendation](#)
 - [Configuring Oracle Identity Manager Release 11.1.2 or Later](#)
 - [Localizing Field Labels in UI Forms for the Connector](#)
 - [Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x](#)
 - [Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x](#)
 - [Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x](#)
 - [Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x](#)
 - [Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x](#)
- Instructions specific to Oracle Identity Manager release 11.1.2.x have been added in the following sections:
 - [Installing the Connector in Oracle Identity Manager](#)
 - [Configuring the IT Resource for the Target System](#)
 - [Configuring the IT Resource for the Connector Server](#)
 - [Configuring Scheduled Jobs for Oracle Database](#)

- [Configuring the Target System As a Trusted Source](#)
- [Configuring Oracle Identity Manager](#)
- [Configuring Scheduled Jobs for MSSQL](#)
- [Configuring the Target System As a Trusted Source](#)
- [Configuring Oracle Identity Manager](#)
- [Configuring Scheduled Jobs for MySQL](#)
- [Configuring the Target System As a Trusted Source](#)
- [Configuring Oracle Identity Manager](#)
- [Configuring Scheduled Jobs for DB2](#)
- [Configuring the Target System As a Trusted Source](#)
- [Configuring Oracle Identity Manager](#)
- [Configuring Scheduled Jobs for Sybase](#)
- [Configuring Oracle Identity Manager](#)
- [Configuring the IT Resource](#)
- Issues tracked by bug numbers 16713445 and 16482139 have been added to [Known Issues and Workarounds](#).
- Bug 14260504 has been removed from [Known Issues and Workarounds](#) as it is not an issue.

Documentation-Specific Updates in Release 11.1.1.6.0

The following is the documentation-specific update in revision "4" of release 11.1.1.6.0:

- In [Table 1-1](#):
 - the patch numbers for Oracle Identity Manager have been replaced with 14163597 and 13897038.
 - Exadata V2 target system has been included.

Documentation-Specific Updates in Release 11.1.1.5.0

There are no documentation-specific updates in this release.

1

About the Connector

The Database User Management connector integrates Oracle Identity Manager with database user management tables in Oracle Database, Microsoft SQL Server, MySQL, IBM DB2, and Sybase.

This chapter contains the following sections:

- [Introduction to the Database User Management Connector](#)
- [Certified Components](#)
- [Usage Recommendation](#)
- [Certified Languages](#)
- [Connector Architecture](#)
- [Features of the Connector](#)
- [Roadmap for Deploying and Using the Connector](#)

1.1 Introduction to the Database User Management Connector

Oracle Identity Manager automates access rights management, security, and provisioning of IT resources. Oracle Identity Manager connectors are used to integrate Oracle Identity Manager with third-party applications. This guide discusses the procedure to deploy the connector that is used to integrate Oracle Identity Manager with database user management tables in the following target systems:

- Oracle Database

In Oracle Database, the Login and User entities are treated as a single entity. In this guide, that entity is referred to as the Login entity.

- Microsoft SQL Server
- MySQL
- DB2
- Sybase

In Microsoft SQL Server and Sybase, access entities can be divided into the following types:

- UserLogin: A login entity is used for authentication purposes.
- User: A user entity is used for authorization or access control purposes.

Microsoft SQL Server and Sybase treat these entities as parent (Login) and child (User) elements. In Oracle Identity Manager, these entities are treated as separate, independent entities. In other words, the connector provides login provisioning as well as user provisioning features in Microsoft SQL Server and Sybase.

**Note:**

At some places in this guide, database resources such as Oracle, MSSQL, MySQL, DB2, or Sybase have been referred to as the **target system**.

1.2 Certified Components

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

Table 1-1 Certified Components

Component	Requirement
Oracle Identity Manager	<p>You can use one of the following releases of Oracle Identity Manager:</p> <ul style="list-style-type: none"> • Oracle Identity Governance 12c (12.2.1.4.0) • Oracle Identity Governance 12c (12.2.1.3.0) • Oracle Identity Manager release 11g PS1 (11.1.1.5.3) BP03 and any later BP in this release track <p>If you are using Oracle Identity Manager 11.1.1.5.3, then you must download and apply the patches 14163597 and 13897038. To download a patch, sign in to My Oracle Support and search for the patch number on the Patches and Updates page at:</p> <p>https://support.oracle.com/</p> <p>Note: In this guide, Oracle Identity Manager release 11.1.1 has been used to denote Oracle Identity Manager release 11.1.1.5 BP03, and future releases in the 11.1.1 series that the connector will support.</p> <ul style="list-style-type: none"> • Oracle Identity Manager 11g Release 2 (11.1.2.0.4) and any later BP in this release track • Oracle Identity Manager 11g Release 2 PS3 (11.1.2.3.0)
Target systems	<p>The target system can be any one of the following:</p> <ul style="list-style-type: none"> • Exadata V2 • Oracle9i Database • Oracle Database 10g and 11g as either single database or Oracle RAC implementation • Oracle Database 12c as single database, pluggable database (PDB), or Oracle RAC implementation • Oracle Database 19c or 18c as a single database, pluggable database (PDB), or Oracle RAC implementation • Microsoft SQL Server 2005, 2008, 2012, 2014, 2016, 2017, 2019 • MySQL 5.x • IBM DB2 UDB 9.x • Sybase 15.x
Connector Server	11.1.2.1.0
Connector Server JDK	JDK 1.6 or later, or JRockit JDK 1.6 or later.

Table 1-1 (Cont.) Certified Components

Component	Requirement
Target system user account	Depending on the target system, one of the following user accounts is used by Oracle Identity Manager to perform reconciliation and provisioning operations on the target system: <ul style="list-style-type: none">• For Oracle Database: <code>sys as sysdba</code>, or <code>system</code>• For Microsoft SQL Server: <code>sa</code> (administrator)• For MySQL: <code>root</code>• For DB2: <code>db2admin</code>• For Sybase: <code>sa</code> (administrator)

1.3 Usage Recommendation

Depending on the Oracle Identity Manager version that you are using, you must deploy and use one of the following connectors:

- If you are using an Oracle Identity Manager release that is later than release 9.1.0.2 and earlier than Oracle Identity Manager 11g Release 1 (11.1.1.5.3), then you must use the 9.1.x version of this connector.
- If you are using Oracle Identity Manager 11g Release 1 (11.1.1.5.3) and any later BP in this release track, Oracle Identity Manager 11g Release 2 (11.1.2.0.4) and any later BP in this release track, or Oracle Identity Manager 11g Release 2 PS3 (11.1.2.3.0) and any later BP in this release track, then you must use the latest 11.1.1.x version of this connector.
- If you are using Microsoft SQL Server 2000 as the target system, then you must use the 9.1.x version of this connector, irrespective of the Oracle Identity Manager release you are using.

1.4 Certified Languages

The connector supports the following languages:

- Arabic
- Chinese (Simplified)
- Chinese (Traditional)
- Czech
- Danish
- Dutch
- English (UK)
- English (US)
- Finnish
- French
- German
- Greek

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



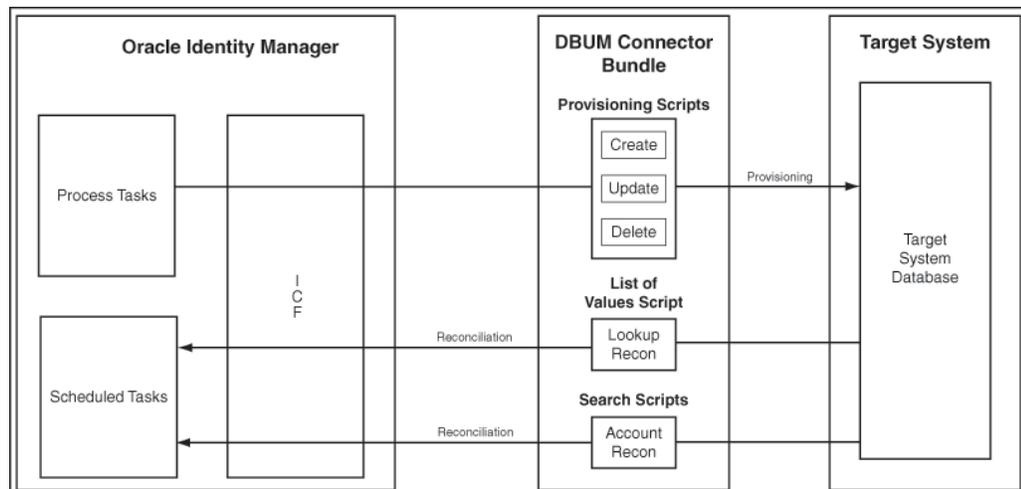
Note:

However, the connector does not support the entry of multibyte characters in some of the fields.

1.5 Connector Architecture

The connector enables management of database accounts through Oracle Identity Manager. [Figure 1-1](#) shows the architecture of the connector.

Figure 1-1 Architecture of the Connector



The Database User Management (DBUM) connector is implemented by using the Identity Connector Framework (ICF). The ICF is a component that provides basic reconciliation and provisioning operations that are common to all Oracle Identity Manager connectors. In addition, ICF provides common features that developers would otherwise need to implement on their own, such as connection pooling, buffering, time outs, and filtering. The ICF is shipped along with Oracle Identity Manager. Therefore, you need not configure or modify the ICF.

See Also:

Understanding the Identity Connector Framework in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for more information about the ICF

The out of the box (OOB) connectors are provided with the certified scripts for the certified targets such as Oracle, MSSQL, MySQL, DB2, and Sybase. If the connector is customized for a database other than the certified ones, then you need to manually add scripts for the new database.

All DBUM operations are performed by executing SQL Scripts or by calling Stored Procedures (Procs). These scripts and stored procs are externalized in different files in the connector bundle and can also be customized. IT resource, that has the configuration lookup parameter contains the bundle key. This bundle key is made of bundle name, bundle version, connector name, and is used for loading the bundle.

The following are the three categories of scripts that are stored in the connector bundle:

Script	Description
Provisioning.queries	This script is used for Create, Update, or Delete operations.
LoVSearch.queries	This script is used for lookup reconciliation. It contains the set of possible values for certain fields such as profile, privileges, roles, and tablespaces.
Search.queries	This script is used for full or incremental or delete reconciliation. You can also perform account and group search with various conditions using this script.

Depending on the query invoked, ExecutionHandler executes the queries. There are two different handlers `SQLExecutionHandler` and `StoredProcExecutionHandler` which extends `ExecutionHandler`.

Depending on the type of Query, corresponding ExecutionHandler is invoked. `StoredProcExecutionHandler` is used for operations in MSSQL. The following is the example used for searching users:

```
USER_DATA_QUERY {
  Query="CALL sp_helpuser({__UID__})"
  QueryType="StoredProc"
  Parameters=["__UID__": "Type:String, Direction:IN",
    "defaultDatabase": "Type:String, Direction:OUT, ColName:DefDBName",
    "loginName": "Type:String, Direction:OUT, ColName:LoginName",
    "roles~DBRole~__NAME__": "Type:String, Direction:OUT, ColName:RoleName"]
  QueryExtensions=[]
}
```

SQL queries are categorized into Data Definition Language (DDL) and Data Manipulation Language (DML) queries. DDL queries are used for CREATE, REVOKE, GRANT, ALTER, and so on, where as DML queries are used for UPDATE, INSERT, and so on.

The DDL queries are executed as regular statements. The following is the example for DDL statement used for the create operation:

```
Statement stmt = null;
    try {
        stmt = _dbConnection.getConnection().createStatement();
        stmt.execute(sqlScript);
    }
```

The DML queries are executed as prepared statements. The following is the example for DML statement used for the update operation:

```
PreparedStatement st = null;
    try {
        st = conn.prepareStatement(sqlScript);
        setParams(st, Arrays.asList(params));
        return st.executeUpdate();
    }
```

The information about the connector bundle is stored in the manifest file. This file contains the connector definition, which gives the information about the connector bundle framework version, connector bundle name, and connector bundle version. The following is the example of the connector definition:

```
org.identityconnectors.dbum.1.0.1116.jar
```

In this example:

org.identityconnectors: refers to connector bundle framework

dbum: refers to the connector bundle name

1.0.1116.jar: refers to the connector bundle version

The connector definition is required to identify the connector bundle. If you are deploying the multiple versions of the target, then you need to change the connector version and redeploy it. You can do it by changing the connector version in the connector bundle present in the manifest file.

See Also:

The following sections for more details about lookup definitions for each database:

- [Lookup Definitions Used During Connector Operations for Oracle Database](#)
- [Lookup Definitions for MSSQL](#)
- [Lookup Definitions for MySQL](#)
- [Lookup Definitions for DB2](#)
- [Lookup Definitions for Sybase](#)

1.6 Features of the Connector

The following are features of the connector:

- [Mapping Standard and Custom Attributes for Reconciliation and Provisioning](#)
- [Predefined and Custom Reconciliation Queries](#)
- [Predefined and Custom Provisioning Queries](#)
- [Support for Creating Global and External Users In Oracle Database](#)
- [Full and Incremental Reconciliation](#)
- [Limited \(Filtered\) Reconciliation](#)
- [Batched Reconciliation](#)
- [Specifying Accounts to Be Excluded from Reconciliation and Provisioning Operations](#)
- [Connection Pooling](#)
- [Support for Connector Server](#)
- [Support for Creating Connector Copies](#)
- [Transformation and Validation of Account Data](#)
- [Support for Reconciling Data About Deleted Entities](#)
- [Separate Scheduled Jobs for Reconciliation of Users, Logins, and Deleted Login Entities](#)
- [Support for SSL Communication Between the Target System and Oracle Identity Manager](#)
- [Support for Managing Authorization to Oracle Database Vault Realms](#)
- [Support for Configuring the Connector for Enterprise User Security](#)

1.6.1 Mapping Standard and Custom Attributes for Reconciliation and Provisioning

You can create mappings for single-valued and multivalued target system attributes that are not included in the list of default attribute mappings. These attributes can be part of the standard set of attributes provided by the target system or custom attributes that you add on the target system.

See the following sections for more information:

- [Extending the Connector for Oracle Database](#)
- [Extending the Connector for MSSQL](#)
- [Extending the Connector for MySQL](#)
- [Extending the Connector for DB2](#)
- [Extending the Connector for Sybase](#)

1.6.2 Predefined and Custom Reconciliation Queries

Reconciliation involves running a SQL query or stored procedure on the target system database to fetch the required user account records to Oracle Identity Manager.

The connector provides predefined SQL queries and stored procedures that enable you to reconcile user data from the target system. These predefined SQL queries and stored procedures are stored in the Search.queries file in the connector bundle.

You can modify these SQL queries or stored procedures. In addition, you can add your own SQL queries or stored procedures for reconciliation.

See the following sections for more information:

- [Syntax of Reconciliation Queries for Oracle Database](#)
- [Syntax of Reconciliation Queries for MSSQL](#)
- [Syntax of Reconciliation Queries for MySQL Database](#)
- [Syntax of Reconciliation Queries for DB2](#)
- [Syntax of Reconciliation Queries for Sybase Database](#)

1.6.3 Predefined and Custom Provisioning Queries

Provisioning involves running a SQL query or stored procedure such as CREATE USER, ALTER USER, and DROP USER to perform Create User and Update user operations on the target system through Oracle Identity Manager.

The connector provides predefined queries that enable you to perform provisioning operations such as create, enable, and update target system accounts. These predefined SQL queries and stored procedures are stored in the Provisioning.queries file in the connector bundle.

You can modify and use any of the predefined provisioning queries. In addition, you can create your own provisioning queries. See the following sections for more information:

- [Syntax of Provisioning Queries for Oracle Database](#)
- [Syntax of Provisioning Queries for MSSQL](#)
- [Syntax of Provisioning Queries for MySQL Database](#)
- [Syntax of Provisioning Queries for DB2](#)
- [Syntax of Provisioning Queries for Sybase Database](#)

1.6.4 Support for Creating Global and External Users In Oracle Database

A local database user is a user who can be authenticated using a password stored in the database. In addition to support for local database users, the connector can also be used to work with the following types of users in Oracle Database:

- **Global users:** These are database users who must be authorized by an enterprise directory service such as Oracle Internet Directory.
- **External users:** These are database users who must be authenticated by an external service, such as an operating system or a third-party service such as Kerberos.

See the following sections for more information about local, global, and external user authentication types:

- [Support for Configuring the Connector for Enterprise User Security](#)
- [Guidelines on Performing Provisioning Operations for Oracle Database](#)

1.6.5 Full and Incremental Reconciliation

After you deploy the connector, you can perform full reconciliation to bring all existing user data from the target system to Oracle Identity Manager. After the first full reconciliation run, you can configure your connector for incremental reconciliation. In incremental reconciliation, only records that are added or modified after the last reconciliation run are fetched into Oracle Identity Manager.

See the following sections for more information:

- [Reconciliation from Oracle Database](#)
- [Reconciliation from MSSQL](#)
- [Reconciliation from MySQL](#)
- [Reconciliation from DB2](#)
- [Reconciliation from Sybase](#)

1.6.6 Limited (Filtered) Reconciliation

ICF filter performs the limited reconciliation and the records are fetched into Oracle Identity Manager during a reconciliation run. The ICF filters are translated to WHERE clause and applied in the Search query.

See the following sections for more information:

- [Performing Limited Reconciliation from Oracle Database](#)
- [Performing Limited Reconciliation from MSSQL](#)
- [Performing Limited Reconciliation from MySQL](#)
- [Performing Limited Reconciliation from DB2](#)

1.6.7 Batched Reconciliation

You can break down a reconciliation run into batches by specifying the number of records that must be included in each batch and the query that must be used to perform batched reconciliation.



Note:

Microsoft SQL Server uses stored procedures to perform reconciliation. Therefore, the connector does not support batched reconciliation.

See the following sections for more information:

- [Performing Batched Reconciliation from Oracle Database](#)
- [Performing Batched Reconciliation from MySQL](#)
- [Performing Batched Reconciliation from DB2](#)

1.6.8 Specifying Accounts to Be Excluded from Reconciliation and Provisioning Operations

You can specify a list of target system accounts that must be excluded from all reconciliation and provisioning operations. Accounts whose users attributes you specify in the exclusion list are not affected by reconciliation and provisioning operations.

See the following sections for more information:

- [Configuring Resource Exclusion Lists for Oracle Database](#)
- [Configuring Resource Exclusion Lists for MSSQL](#)
- [Configuring Resource Exclusion Lists for MySQL](#)
- [Configuring Resource Exclusion Lists for DB2](#)
- [Configuring Resource Exclusion Lists for Sybase](#)

1.6.9 Connection Pooling

A connection pool is a cache of objects that represent physical connections to the target. Oracle Identity Manager connectors can use these connections to communicate with target systems. At run time, the application requests a connection from the pool. If a connection is available, then the connector uses it and then returns it to the pool. A connection returned to the pool can again be requested for and used by the connector for another operation. By enabling the reuse of connections, the connection pool helps reduce connection creation overheads like network latency, memory allocation, and authentication.

One connection pool is created for each IT resource. For example, if you have three IT resources for three installations of the target system, then three connection pools will be created, one for each target system installation.

The configuration properties of the connection pool are part of the lookup configuration. [Setting up the Lookup Definition for Connection Pooling](#) provides information about setting up the connection pool.

1.6.10 Support for Connector Server

Connector Server is a component provided by ICF. By using one or more connector servers, the connector architecture permits your application to communicate with externally deployed bundles. In other words, a connector server enables remote execution of an Oracle Identity Manager connector.

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

See [Installing and Configuring the Connector Server](#) for more information.

1.6.11 Support for Creating Connector Copies

You can configure this connector for multiple installations of your target system by creating copies of connector objects such as lookup definitions, resource objects, and process forms.

See the following sections for more information:

- [About Configuring the Connector for Multiple Installations of Oracle Database](#)
- [About Configuring the Connector for Multiple Installations of MSSQL](#)
- [About Configuring the Connector for Multiple Installations of MySQL](#)
- [About Configuring the Connector for Multiple Installations of DB2](#)
- [About Configuring the Connector for Multiple Installations of Sybase](#)

1.6.12 Transformation and Validation of Account Data

You can configure validation of account data that is brought into or sent from Oracle Identity Manager during reconciliation and provisioning. In addition, you can configure transformation of account data that is brought into Oracle Identity Manager during reconciliation. The following sections provide more information:

For Oracle Database:

- [Configuring Validation of Data During Reconciliation and Provisioning for Oracle Database](#)
- [Configuring Transformation of Data During User Reconciliation for Oracle Database](#)

For MSSQL:

- [Configuring Validation of Data During Reconciliation and Provisioning for MSSQL](#)
- [Configuring Transformation of Data During User Reconciliation for MSSQL](#)

For MySQL:

- [Configuring Validation of Data During Reconciliation and Provisioning for MySQL](#)
- [Configuring Transformation of Data During User Reconciliation for MySQL](#)

For DB2:

- [Configuring Validation of Data During Reconciliation and Provisioning for DB2](#)
- [Configuring Transformation of Data During User Reconciliation for DB2](#)

For Sybase:

- [Configuring Validation of Data During Reconciliation and Provisioning for Sybase](#)
- [Configuring Transformation of Data During User Reconciliation for Sybase](#)

1.6.13 Support for Reconciling Data About Deleted Entities

You can reconcile data about login entities that have been deleted on the target system that has been configured as a trusted source or target resource.

After the records are fetched in to Oracle Identity Manager, depending on whether you have configured your target system as a target resource or trusted source, the records are

compared with existing OIM Users or database resources provisioned to existing OIM Users. The unmatched accounts are revoked/removed from Oracle Identity Manager.

1.6.14 Separate Scheduled Jobs for Reconciliation of Users, Logins, and Deleted Login Entities

You can reconcile data about users, logins, or deleted login entities from a target system that is configured as a trusted source or target resource. Depending on the target system that you are using, the mode in which it is configured, and the type of data that you want to reconcile, separate scheduled jobs have been created.

See the following sections for more information:

- [Configuring Scheduled Jobs for Oracle Database](#)
- [Configuring Scheduled Jobs for MSSQL](#)
- [Configuring Scheduled Jobs for MySQL](#)
- [Configuring Scheduled Jobs for DB2](#)
- [Configuring Scheduled Jobs for Sybase](#)

1.6.15 Support for SSL Communication Between the Target System and Oracle Identity Manager

You can configure SSL to secure communication between Oracle Identity Manager and the target system. See the following sections for more information:

- [Configuring Secure Communication Between Oracle Database and Oracle Identity Manager](#)
- [Configuring Secure Communication Between MSSQL and Oracle Identity Manager](#)
- [Configuring Secure Communication Between MySQL and Oracle Identity Manager](#)
- [Configuring Secure Communication Between DB2 and Oracle Identity Manager](#)
- [Configuring Secure Communication Between Sybase and Oracle Identity Manager](#)

1.6.16 Support for Managing Authorization to Oracle Database Vault Realms

Oracle Database Vault restricts access to specific areas in an Oracle Database from any user, including users who have administrative access. For example, you can restrict administrative access to employee salaries, customer medical records, or other sensitive information. This enables you to apply fine-grained access control to your sensitive data in a variety of ways. It hardens your Oracle Database instance and enforces industry standard best practices in terms of separating duties from users with administrative access. Most importantly, it protects data from super-privileged users but still allows them to manage the Oracle Database installation.

With Oracle Database Vault, you can address business requirements such as protecting against insider threats, meeting regulatory compliance requirements, and enforcing separation of duty.

You configure Oracle Database Vault to manage the security of an individual Oracle Database instance. You can install Oracle Database Vault on standalone Oracle Database installations, in multiple Oracle homes, and in Oracle Real Application Clusters (Oracle RAC) environments.

In Oracle Database installations on which Oracle Database Vault is installed, the connector can be used to grant and manage authorization to Oracle Database Vault realms. The connector treats access to Oracle Database Vault realms as an entitlement. You can use the connector to provision database users with access to multiple realms with different levels of access.

Because Oracle Identity Manager is an enterprise application for managing user accounts and access to entitlements, the connector does not support management of the following:

- Realms
- Command rules and rule sets
- Factors
- Secure Application Roles

See [Creating the Administrator Account on Oracle Database Vault](#) for more information.

1.6.17 Support for Configuring the Connector for Enterprise User Security

Oracle Enterprise User Security addresses user, administrative, and security challenges by using the identity management services supplied by Oracle Internet Directory, an LDAP-compliant directory service. You must use either Oracle Identity Manager LDAP connectors or some other means to create the user in the LDAP-compliant directory. Enterprise users are provisioned and managed centrally in an LDAP-compliant directory, such as Oracle Internet Directory, for database access. Enterprise users have a unique identity in the directory called the distinguished name (DN). When enterprise users log on to a database, the database authenticates those users by using their DN.

In Oracle Database installations configured with Oracle Enterprise User Security, the connector supports the creation of password, global, and external authenticated users for a target system account (login or user). Depending on the authentication type, you need to make some changes in the process form. If the authentication type is password or external, then you must remove the other authentication fields from the process form. If the authentication type is global, then you must make the following changes in the process form:

- Remove the password field as it is not required for the global authentication.
- Set authentication type from default to Global and make it read-only. This will prevent user from choosing the other authentication type.
- In Global DN, provide the unique ID.

You can use the connector to create and manage accounts of these enterprise users on the target database.

1.7 Roadmap for Deploying and Using the Connector

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

- [Deploying the Connector](#) describes procedures that you must perform on Oracle Identity Manager and the target system during each stage of connector deployment.

- [Using and Extending the Connector for Oracle Database](#) describes guidelines on using the connector and the procedure to configure reconciliation runs and perform provisioning operations for Oracle database.
- [Using and Extending the Connector for MSSQL](#) describes guidelines on using the connector and the procedure to configure reconciliation runs and perform provisioning operations for MSSQL database.
- [Using and Extending the Connector for MySQL](#) describes guidelines on using the connector and the procedure to configure reconciliation runs and perform provisioning operations for MySQL database.
- [Using and Extending the Connector for DB2](#) describes guidelines on using the connector and the procedure to configure reconciliation runs and perform provisioning operations for DB2 database.
- [Using and Extending the Connector for Sybase](#) describes guidelines on using the connector and the procedure to configure reconciliation runs and perform provisioning operations for Sybase database.
- [Configuring the Connector for a JDBC-Based Database](#) describes procedures for creating the connector for the JDBC-based database.
- [Testing the Connector](#) describes procedures to test the connector.
- [Known Issues and Workarounds](#) lists known issues associated with this release of the connector.

2

Deploying the Connector

Preinstallation information is divided across the following sections:

- [Preinstallation](#)
- [Installation](#)
- [Postinstallation](#)
- [Upgrading the Connector](#)
- [Postcloning Steps](#)
- [Postcloning Configuration for User Accounts](#)

2.1 Preinstallation

Preinstallation information is divided across the following sections:

- [Files and Directories on the Installation Media](#)
- [Preinstallation on Microsoft SQL Server](#)
- [Creating a Target System User Account for Connector Operations.](#)

2.1.1 Files and Directories on the Installation Media

[Table 2-1](#) describes the files and directories on the installation media.

Table 2-1 Files and Directories on the Installation Media

File in the Installation Media Directory	Description
File in the bundle directory: org.identityconnectors.dbum-1.0.1116.jar	This file contains connector code, SQL queries, and stored procedures that are used for provisioning and reconciliation.
Files in the configuration directory: DBUM-Oracle-CI.xml DBUM-MSSQL-CI.xml DBUM-MySQL-CI.xml DBUM-DB2-CI.xml DBUM-Sybase-CI.xml	This directory contains the configuration files that are used by the Connector Installer during installation of the connector for a particular target system.
Files in the javadoc directory	This directory contains information about the Java APIs used by the connector.
File in the lib directory: DBUM-oim-integration.jar	This JAR file contains the class files that are used during reconciliation and provisioning operations. During connector installation, this file is copied to the Oracle Identity Manager database.

Table 2-1 (Cont.) Files and Directories on the Installation Media

File in the Installation Media Directory	Description
Files in the resources directory	Each of these resource bundles contains language-specific information that is used by the connector. During connector deployment, this file is copied to the Oracle Identity Manager database location. Note: A resource bundle is a file containing localized versions of the text strings that include GUI element labels and messages.
Files in the test directory: config\oracleconfig.properties config\mssqlconfig.properties config\mysqlconfig.properties config\db2config.properties config\sybaseconfig.properties lib\DBUMTest.jar scripts\DBUMProvisioningTester.bat scripts\DBUMProvisioningTester.sh thirdparty (folder)	This directory contains the files for testing the connector.
Files in the upgrade directory: PostUpgradeScriptOracleDBUM.sql PostUpgradeScriptMSSQLDBUM.sql PostUpgradeScriptMySQLDBUM.sql PostUpgradeScriptDB2DBUM.sql PostUpgradeScriptSybaseDBUM.sql	This directory contains the scripts for performing the post-upgrade operations.
Files in the xml directory: DBUserManagement-Oracle-ConnectorConfig.xml DBUserManagement-Oracle-Datasets.xml DBUserManagement-MSSQL-ConnectorConfig.xml DBUserManagement-MSSQL-Datasets.xml DBUserManagement-MySQL-ConnectorConfig.xml DBUserManagement-MySQL-Datasets.xml DBUserManagement-DB2-ConnectorConfig.xml DBUserManagement-DB2-Datasets.xml DBUserManagement-Sybase-ConnectorConfig.xml DBUserManagement-Sybase-Datasets.xml Note: The dataset XML files are applicable only if you are using Oracle Identity Manager release 11.1.1.x.	This directory contains configuration (target and trusted) XML files and dataset XML files specific to the target system. The configuration XML files contain definitions for the various connector objects, such as resource objects and scheduled jobs, where as the dataset XML files contain datasets for the request based operations. <ul style="list-style-type: none"> IT resource type Process form for each login entity Process form for each user entity Process tasks for each login entity Process tasks for each user entity Resource objects for each login entity Resource objects for each user entity Provisioning Processes for each login entity

2.1.2 Preinstallation on Microsoft SQL Server

If you are using Microsoft SQL Server, then verify the preinstallation requirements by performing the following steps before deploying the connector:

- The target database in which users are to be created exists in the target Microsoft SQL Server installation.
- The TCP/IP port is enabled. The default port is 1433.
To enable the TCP/IP port:
 1. Open the Microsoft SQL Server Configuration Manager.
 2. Click **SQL Server Network Configuration**.
 3. Click **Protocols for MSSQLSERVER**.
 4. In the right frame, right-click **TCP/IP** and then click **Enable**.
- The TCP/IP port is not the only port enabled. Ports other than the TCP/IP port must also be enabled.
- Mixed mode authentication is enabled.
- The TCP/IP port is not blocked by a firewall.

2.1.3 Creating a Target System User Account for Connector Operations

Oracle Identity Manager requires a target system user account to access the target system during reconciliation and provisioning operations. You provide the credentials of this user account while performing the procedure described in [Configuring the IT Resource for the Target System](#).

To create a target system user account for connector operations, depending on the target system you are using, create a user in your target system and assign the mentioned permissions and roles to the user.



See Also:

Target system documentation for detailed information about creating the user

- For MSSQL:
 1. Create Login using the following query:

```
Create LOGIN serviceuser with PASSWORD='password'  
, DEFAULT_DATABASE =DBname  
GO
```
 2. Create a user using the following query:

```
USE DBname;  
Create USER serviceuser with LOGIN serviceuser;  
GO
```
 3. Assign the following permissions and roles to the created user:
 - ALTER ROLE db_datawriter ADD MEMBER serviceuser;
 - ALTER ROLE db_datareader ADD MEMBER serviceuser;
 - ALTER ROLE db_accessadmin ADD MEMBER serviceuser;
 - ALTER ROLE db_owner ADD MEMBER serviceuser;
 - exec sp_addsrvrolemember 'serviceuser', 'securityadmin';

- For Oracle Database:

1. Create Login using the following query:

```
CREATE USER serviceuser IDENTIFIED BY password
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp QUOTA UNLIMITED ON users;
```

2. Assign the following permissions and roles to the created user:

- GRANT CONNECT TO serviceuser;
- GRANT SELECT on dba_role_privs TO serviceuser;
- GRANT SELECT on dba_sys_privs TO serviceuser;
- GRANT SELECT on dba_ts_quotas TO serviceuser;
- GRANT SELECT on dba_tablespaces TO serviceuser;
- GRANT SELECT on dba_users TO serviceuser;
- GRANT CREATE USER TO serviceuser;
- GRANT ALTER ANY TABLE TO serviceuser;
- GRANT GRANT ANY PRIVILEGE TO serviceuser;
- GRANT GRANT ANY ROLE TO serviceuser;
- GRANT DROP USER TO serviceuser;
- GRANT SELECT on dba_roles TO serviceuser;
- GRANT SELECT ON dba_profiles TO serviceuser;
- GRANT ALTER USER TO serviceuser;
- GRANT CREATE ANY TABLE TO serviceuser;
- GRANT DROP ANY TABLE TO serviceuser;
- GRANT CREATE ANY PROCEDURE TO serviceuser;
- REVOKE DROP ANY PROCEDURE TO serviceuser;

- For MySQL:

1. Create a user using the following query:

```
CREATE USER serviceuser IDENTIFIED BY 'password';
```

2. Assign the following permissions and roles to the created user using the following query:

```
GRANT, SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER ON *.* TO
'serviceuser';
```

- For DB2:

1. Create a User 'serviceuser' at the OS level.

2. Assign the following permissions and roles to the created user:

- GRANT SELECT on TABLE syscat.schemata TO serviceuser
- GRANT SELECT on TABLE syscat.tablespaces TO serviceuser
- GRANT CREATEIN,DROPIN,ALTERIN ON SCHEMA 'SCHEMA_NAME' TO serviceuser

```
– GRANT  
CONNECT, BINDADD, DBADM, CREATETAB, CREATE_NOT_FENCED_ROUTINE, IMPLICIT_SC  
HEMA, LOAD, CREATE_EXTERNAL_ROUTINE, QUIESCE_CONNECT ON DATABASE TO  
serviceuser
```

- For Sybase:

1. Create login using the following query:

```
sp_addlogin serviceuser, password
```

2. Create a user using the following query:

```
sp_adduser serviceuser
```

3. Assign the following permissions and roles to the created user using the following queries:

```
– GRANT ROLE sso_role TO serviceuser
```

```
– GRANT ROLE oper_role TO serviceuser
```

2.2 Installation

The following topics provide details on installing the Database User Management Connector:

- [Understanding the Installation of the Connector](#)
- [Installing the Connector in Oracle Identity Manager](#)
- [Deploying the Connector Bundle in a Connector Server](#)

2.2.1 Understanding the Installation of the Connector

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

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

- To run the connector code locally in Oracle Identity Manager, perform the procedure described in [Installing the Connector in Oracle Identity Manager](#)
- To run the connector code remotely in a Connector Server, perform the procedures described in [Installing the Connector in Oracle Identity Manager](#) and [Deploying the Connector Bundle in a Connector Server](#).

 **Note:**

- In this guide, the term **Connector Installer** has been used to refer to the Connector Installer feature of the Oracle Identity Manager Administrative and User Console.
- If you are performing the installation for the second time, then the connector bundle has to be downloaded and the new thirdparty has to be added.
- For Oracle Identity Manager hosted on a Microsoft Windows computer, if you have a previously installed connector, then you must extract the connector bundle again before installing a new connector.
- Database drivers are not needed as they are already loaded for Oracle Identity Manager operations. However, if you want to use the connector with previous versions of database (such as Oracle 9i), then you must use a remote connector server.

2.2.2 Installing the Connector in Oracle Identity Manager

In this scenario, you install the connector in Oracle Identity Manager using the Connector Installer.

 **Note:**

in this guide, the term **Connector Installer** has been used to refer to the Connector Installer feature of the Oracle Identity Manager Administrative and User Console.

To run the Connector Installer:

1. Copy the contents of the connector installation media directory into the following directory:
OIM_HOME/server/ConnectorDefaultDirectory
2. Copy the third party jars to target systems in the *ConnectorDefaultDirectory/targetsystems-lib/DBUM-11.1.1.6.0* directory.

 **Note:**

If the target is Oracle database, then no driver jar is needed. For other target systems, the following third party jar has to be copied:

- For MSSQL, copy sqljdbc4.jar.
- For MySQL, copy mysql-connector-java-5.1.20-bin.jar.
- For DB2, copy db2jcc.jar.
- For Sybase, copy jconn4.jar.

3. If you are using Oracle Identity Manager release 11.1.1.x:
 - a. Log in to the Administrative and User Console.
 - b. On the Welcome to Identity Manager Advanced Administration page, in the System Management region, click **Manage Connector**.
4. If you are using Oracle Identity Manager release 11.1.2.x or later:
 - a. Log in to Oracle Identity System Administration.
 - b. In the left pane, under System Management, click **Manage Connector**.
5. In the Manage Connector page, click **Install**.
6. The Connector List displays the names and release numbers of connectors whose installation files you copy into the default connector installation directory in Step 1.

You can select one of the following options:

- For Oracle: **Oracle DB User Management 11.1.1.8.0**
- For MSSQL: **MSSQL DB User Management 11.1.1.8.0**
- For MySQL: **MySQL DB User Management 11.1.1.8.0**
- For DB2: **DB2 DB User Management 11.1.1.8.0**
- For Sybase: **Sybase DB User Management 11.1.1.8.0**

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

- a. In the **Alternative Directory** field, enter the full path and name of that directory.
 - b. To repopulate the list of connectors in the Connector List options, click **Refresh**.
 - c. From the Connector List options, select:
 - For Oracle: **Oracle DB User Management 11.1.1.8.0**
 - For MSSQL: **MSSQL DB User Management 11.1.1.8.0**
 - For MySQL: **MySQL DB User Management 11.1.1.8.0**
 - For DB2: **DB2 DB User Management 11.1.1.8.0**
 - For Sybase: **Sybase DB User Management 11.1.1.8.0**
7. Click **Load**.
 8. To start the installation process, click **Continue**.

The following tasks are performed in sequence:

- a. Configuration of connector libraries

- b. Import of the connector XML files (by using the Deployment Manager)
- c. Compilation of tasks

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

- Retry the installation by clicking **Retry**.
 - Cancel the installation and begin again from Step 1.
9. If all three tasks of the connector installation process are successful, then a message indicating successful installation is displayed. In addition, a list of the steps that you must perform after the installation is displayed. These steps are as follows:
 - a. Ensuring that the prerequisites for using the connector are addressed

 **Note:**

At this stage, run the Oracle Identity Manager PurgeCache utility to load the server cache with content from the connector resource bundle in order to view the list of prerequisites. See [Clearing Content Related to the Connector Resource Bundles from the Server Cache](#) for information about running the PurgeCache utility.

There are no prerequisites for some predefined connectors.

- b. Configuring an IT resource for the connector

Record the name of the IT resource displayed on this page. See [Configuring the IT Resource for the Target System](#) for the IT Resource details.

- c. Configuring the scheduled jobs that are created when you installed the connector

Record the names of the scheduled jobs displayed on this page. See [Configuring Scheduled Jobs for Oracle Database](#) for a sample procedure to configure these scheduled jobs. There are similar sections for other databases in this guide.

When you run the Connector Installer, it copies the connector files and external code files to destination directories on the Oracle Identity Manager host computer. These files are listed in [Table 2-1](#).

2.2.3 Deploying the Connector Bundle in a Connector Server

You can deploy the Database User Management connector either locally in Oracle Identity Manager or remotely in the Connector Server. A *connector server* is an application that enables remote execution of an Identity Connector, such as the DBUM connector.

 **Note:**

- To deploy the connector bundle remotely in a Connector Server, you must first deploy the connector in Oracle Identity Manager, as described in [Installing the Connector in Oracle Identity Manager](#).
- See [Configuring the IT Resource for the Connector Server](#) for related information.

This procedure can be divided into the following stages:

- [Installing and Configuring the Connector Server](#)
- [Running the Connector Server](#)
- [Installing the Connector on the Connector Server](#)

2.2.3.1 Installing and Configuring the Connector Server

Connector servers are available in two implementations:

- As a .Net implementation that is used by Identity Connectors implemented in .Net
- As a Java Connector Server implementation that is used by Java-based Identity Connectors

The DBUM connector is implemented in Java, so you can deploy this connector to a Java Connector Server.

Use the following steps to install and configure the Java Connector Server:

 **Note:**

Before you deploy the Java Connector Server, ensure that you install the JDK or JRE on the same computer where you are installing the Java Connector Server and that your `JAVA_HOME` or `JRE_HOME` environment variable points to this installation.

1. Create a new directory on the computer where you want to install the Java Connector Server.

 **Note:**

In this guide, `CONNECTOR_SERVER_HOME` represents this directory.

2. Unzip the Java Connector Server package in the new directory created in Step 1. You can download the Java Connector Server package from the Oracle Technology Network.
3. Open the `ConnectorServer.properties` file located in the `conf` directory. In the `ConnectorServer.properties` file, set the following properties, as required by your deployment.

Property	Description
connectorserver.port	Port on which the Java Connector Server listens for requests. Default is 8763.
connectorserver.bundleDir	Directory where the connector bundles are deployed. Default is <code>bundles</code> .
connectorserver.libDir	Directory in which to place dependent libraries. Default is <code>lib</code> .
connectorserver.usessl	If set to <code>true</code> , the Java Connector Server uses SSL for secure communication. Default is <code>false</code> . If you specify <code>true</code> , use the following options on the command line when you start the Java Connector Server: <ul style="list-style-type: none"> • <code>-Djavax.net.ssl.keyStore</code> • <code>-Djavax.net.ssl.keyStoreType</code> (<i>optional</i>) • <code>-Djavax.net.ssl.keyStorePassword</code>
connectorserver.ifaddress	Bind address. To set this property, uncomment it in the file (if necessary). The bind address can be useful if there are more NICs installed on the computer.
connectorserver.key	Java Connector Server key.

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

 **Note:**

For more information, see [Running the Connector Server](#).

- For all other properties, edit the `ConnectorServer.properties` file manually.
5. The `conf` directory also contains the `logging.properties` file, which you can edit if required by your deployment.

 **Note:**

Oracle Identity Manager has no built-in support for connector servers, so you cannot test your configuration.

2.2.3.2 Running the Connector Server

To run the Java Connector Server, use the `ConnectorServer.bat` script for Windows and use the `ConnectorServer.sh` script for UNIX as follows:

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

The ConnectorServer.bat supports the following options:

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

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

2.2.3.3 Installing the Connector on the Connector Server

See Also:

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

If you need to deploy the DBUM into the Java Connector Server, then follow these steps:

1. Stop the Java Connector Server.

Note:

You can download the necessary Java Connector Server from the Oracle Technology Network web page.

2. Copy the DBUM connector bundle into the Java Connector Server `CONNECTOR_SERVER_HOME\bundles` directory.
3. Copy the DBUM third party libraries to the `CONNECTOR_SERVER_HOME\lib` directory. If multiple versions of the same connector are present, then third party has to be bundled within the connector bundle jar. To do so:
 - a. Create temporary/lib folder and drop third party jars in it.

- b. Update the bundle with the third party jar:

```
jar -uvf org.identityconnectors.dbum-1.0.1116.jar lib/JAR_NAME
```
- c. Remove temporary/lib folder.
- d. Start the Connector Server.

 **Note:**

if there are multiple versions of the same connector bundle, then the third-party JAR should go into bundle instead of the `CONNECTOR_SERVER_HOME/lib` directory.

4. Start the Java Connector Server.

2.3 Postinstallation

Postinstallation steps are divided across the following sections:

- [Postinstallation on Oracle Identity Manager](#)
- [Configuring the IT Resource for the Target System](#)
- [Configuring the Connector to Support Multiple Versions of the Target System](#)
- [Configuring the IT Resource for the Connector Server](#)
- [Enabling Logging for Connector Server](#)

2.3.1 Postinstallation on Oracle Identity Manager

This section discusses the following topics:

- [Configuring the Target System As a Trusted Source](#)
- [Changing to the Required Input Locale](#)
- [Clearing Content Related to the Connector Resource Bundles from the Server Cache](#)
- [Creating the Administrator Account on Oracle Database Vault](#)
- [Setting up the Lookup Definition for Connection Pooling](#)
- [Managing Logging for Oracle Identity Manager](#)
- [Configuring Oracle Identity Manager Release 11.1.2 or Later](#)
- [Localizing Field Labels in UI Forms for the Connector](#)

2.3.1.1 Configuring the Target System As a Trusted Source

 **Note:**

You can skip this section if you do not want to designate the target system as a trusted source for reconciliation.

You can designate the target system as a trusted source or target resource. If you designate the target system as a **trusted source**, then during a reconciliation run:

- For each newly created user on the target system, an OIM User is created.
- Updates made to each user on the target system are propagated to the corresponding OIM User.

To configure trusted source reconciliation, create and configure a new IT resource.



See Also:

[Configuring the IT Resource for the Target System](#) for more information about configuring the IT resource for the target system

1. Create a new IT resource:
 - For Oracle: Oracle DB Trusted of type Oracle DBUM
 - For MSSQL: MSSQL DB Trusted of type MSSQL DBUM
 - For MySQL: MSSQL DB Trusted of type MSSQL DBUM
 - For DB2: DB2 DB Trusted of type DB2 DBUM
 - For Sybase: Sybase DB Trusted of type Sybase DBUM
2. In the Configuration Lookup, update the trusted configuration lookup name to:
 - For Oracle: Lookup.DBUM.Oracle.Configuration.Trusted
 - For MSSQL: Lookup.DBUM.MSSQL.Configuration.Trusted
 - For MySQL: Lookup.DBUM.MySQL.Configuration.Trusted
 - For DB2: Lookup.DBUM.DB2.Configuration.Trusted
 - For Sybase: Lookup.DBUM.Sybase.Configuration.Trusted

2.3.1.2 Changing to the Required Input Locale

Changing to the required input locale (language and country setting) involves installing the required fonts and setting the required input locale.

You may require the assistance of the system administrator to change to the required input locale.

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



Note:

In an Oracle Identity Manager cluster, you must perform these steps on each node of the cluster. Then, restart each node.

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

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

1. In a command window, switch to the *OIM_HOME/server/bin* directory.

 **Note:**

You must perform Step 1 before you perform Step 2. An exception is thrown if you run the command described in Step 2 as follows:

```
OIM_HOME/server/bin/SCRIPT_FILE_NAME
```

2. Enter one of the following commands:

 **Note:**

You can use the PurgeCache utility to purge the cache for any content category. Run `PurgeCache.bat CATEGORY_NAME` on Microsoft Windows or `PurgeCache.sh CATEGORY_NAME` on UNIX. The *CATEGORY_NAME* argument represents the name of the content category that must be purged.

For example, the following commands purge Metadata entries from the server cache:

```
PurgeCache.bat MetaData
```

```
PurgeCache.sh MetaData
```

On Microsoft Windows: `PurgeCache.bat All`

On UNIX: `PurgeCache.sh All`

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

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

In this format:

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

2.3.1.4 Creating the Administrator Account on Oracle Database Vault



Note:

Perform the procedure described in this section only if you have Oracle Database Vault installed and you want to configure the connector for provisioning and reconciling authorization to Oracle Database Vault realms.

You must create an administrator account on Oracle Database Vault. This account is used by the connector for performing reconciliation and provisioning operations on Oracle Database Vault realms.

To create the administrator account on Oracle Database Vault:

1. Log in to Oracle Database Vault as a user with the DV_ACCTMGR privilege.
2. Create the administrator account by running the following command:

```
CREATE USER USERNAME IDENTIFIED BY PASSWORD;
```

3. Log out and then log in as a user with the DV_OWNER privilege.
4. Grant access to Oracle Database Vault and Data Dictionary realms by running the following commands:

```
exec DVSYS.DBMS_MACADM.ADD_AUTH_TO_REALM('Database Vault Account
Management', 'USERNAME', 'Enabled', 1)
exec DVSYS.DBMS_MACADM.ADD_AUTH_TO_REALM('Oracle Data
Dictionary', 'USERNAME', 'Enabled', 1)
```

5. Grant the DV_ADMIN and DV_SECANALYST privileges.
6. Log in as a user with the DV_ACCTMGR privilege.
7. Grant the DV_SECANALYST privilege.
8. Log in as SYS and grant the following privileges (run the command):

```
GRANT ANY OBJECT PRIVILEGE
GRANT ANY PRIVILEGE
GRANT ANY ROLE
UNLIMITED TABLESPACE
with ADMIN OPTION
to USERNAME
```

2.3.1.5 Setting up the Lookup Definition for Connection Pooling

By default, this connector uses the ICF connection pooling. [Table 2-2](#) lists the connection pooling properties, their description, and default values set in ICF:

Table 2-2 Connection Pooling Properties

Property	Description
Pool Max Idle	Maximum number of idle objects in a pool. Default value: 10

Table 2-2 (Cont.) Connection Pooling Properties

Property	Description
Pool Max Size	Maximum number of connections that the pool can create. Default value: 10
Pool Max Wait	Maximum time, in milliseconds, the pool must wait for a free object to make itself available to be consumed for an operation. Default value: 150000
Pool Min Evict Idle Time	Minimum time, in milliseconds, the connector must wait before evicting an idle object. Default value: 120000
Pool Min Idle	Minimum number of idle objects in a pool. Default value: 1

If you want to modify the connection pooling properties to use values that suit requirements in your environment, then:

1. Log in to the Design Console.
2. Expand **Administration**, and then double-click **Lookup Definition**.
3. Search for and open the configuration lookup definition for the target system you are using.
For example, open Lookup.DBUM.Oracle.Configuration for Oracle Database.
4. On the Lookup Code Information tab, click **Add**.
A new row is added.
5. In the **Code Key** column of the new row, enter `Pool Max Idle`.
6. In the **Decode** column of the new row, enter a value corresponding to the Pool Max Idle property.
7. Repeat Steps 4 through 6 for adding each of the connection pooling properties listed in [Table 2-2](#).
8. Click the save icon.

2.3.1.6 Managing Logging for Oracle Identity Manager

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

The following topics provide detailed information about logging:

- [Understanding Log Levels](#)
- [Enabling Logging](#)

2.3.1.6.1 Understanding Log Levels

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

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

- `SEVERE.intValue()+100`
This level enables logging of information about fatal errors.
- `SEVERE`
This level enables logging of information about errors that might allow Oracle Identity Manager to continue running.
- `WARNING`
This level enables logging of information about potentially harmful situations.
- `INFO`
This level enables logging of messages that highlight the progress of the application.
- `CONFIG`
This level enables logging of information about fine-grained events that are useful for debugging.
- `FINE, FINER, FINEST`
These levels enable logging of information about fine-grained events, where `FINEST` logs information about all events.

These message types are mapped to ODL message type and level combinations as shown in [Table 2-3](#).

Table 2-3 Log Levels and ODL Message Type:Level Combinations

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

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

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

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

2.3.1.6.2 Enabling Logging

To enable logging in Oracle WebLogic Server:

1. Edit the `logging.xml` file as follows:

- a. Add the following blocks in the file:

```
<log_handler name='db-um-handler' level=' [LOG_LEVEL] '
class='oracle.core.ojdl.logging.ODLHandlerFactory'>
<property name='logreader:' value='off' />
  <property name='path' value=' [FILE_NAME] ' />
  <property name='format' value='ODL-Text' />
  <property name='useThreadName' value='true' />
  <property name='locale' value='en' />
  <property name='maxFileSize' value='5242880' />
  <property name='maxLogSize' value='52428800' />
  <property name='encoding' value='UTF-8' />
</log_handler>

<logger name="ORG.IDENTITYCONNECTORS.DBUM" level=" [LOG_LEVEL] "
useParentHandlers="false">
  <handler name="db-um-handler" />
  <handler name="console-handler" />
</logger>
```

- b. Replace all occurrences of [LOG_LEVEL] with the ODL message type and level combination that you require. Table 2-3 lists the supported message type and level combinations.

Similarly, replace [FILE_NAME] with the full path and name of the log file in which you want log messages to be recorded.

The following blocks show sample values for [LOG_LEVEL] and [FILE_NAME] :

```
<log_handler name='db-um-handler' level='NOTIFICATION:1'
class='oracle.core.ojdl.logging.ODLHandlerFactory'>
<property name='logreader:' value='off' />
  <property name='path'
value='F:\MyMachine\middleware\user_projects\domains\base_domain1\servers
\oim_server1\logs\oim_server1-diagnostic-1.log' />
  <property name='format' value='ODL-Text' />
  <property name='useThreadName' value='true' />
  <property name='locale' value='en' />
  <property name='maxFileSize' value='5242880' />
  <property name='maxLogSize' value='52428800' />
  <property name='encoding' value='UTF-8' />
</log_handler>

<logger name="oracle.iam.connectors.icfcommon" level="NOTIFICATION:1"
useParentHandlers="false">
  <handler name="db-um-handler" />
</logger>
<logger name="ORG.IDENTITYCONNECTORS.DBUM" level="NOTIFICATION:1"
useParentHandlers="false">
  <handler name="db-um-handler" />
</logger>
```

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

2. Save and close the file.
3. Restart the application server.

2.3.1.7 Configuring Oracle Identity Manager Release 11.1.2 or Later

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

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

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

2.3.1.7.1 Creating and Activating a Sandbox

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

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

2.3.1.7.2 Creating a New UI Form

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

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

2.3.1.7.3 Creating an Application Instance

Create an application instance as follows:

1. In the left pane, under Configuration, click **Application Instances**. The Application Instances page is displayed.
2. From the Actions menu, select **Create**. Alternatively, click **Create** on the toolbar. The Create Application Instance page is displayed.
3. Specify values for the following fields:
 - **Name:** The name of the application instance.
 - **Display Name:** The display name of the application instance.
 - **Description:** A description of the application instance.
 - **Resource Object:** The resource object name. Depending on the target system that you are using, click the search icon next to this Resource Object field to search for and select one of the following:

For Oracle Database: **Oracle DB User**

For MSSQL: **MSSQL DB User Login** or **MSSQL DB User**

For MySQL: **MySQL DB User**

For DB2: **DB2 DB User**

For Sybase: **Sybase DB User**

- **IT Resource Instance:** The IT resource instance name. Depending on the target system that you are using, click the search icon next to this IT Resource Instance field to search for and select one of the following:

For Oracle Database: **Oracle DB**

For MSSQL: **MSSQL DB**

For MySQL: **MySQL DB**

For DB2: **DB2**

For Sybase: **Sybase DB**

- **Form:** Select the form name (created in Step 3).
4. Click Save. The application instance is created.
 5. Publish the application instance to an organization to make the application instance available for requesting and subsequent provisioning to users. See *Managing Organizations Associated With Application Instances in Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed instructions.

**Note:**

Creating Application Instances of *Oracle Fusion Middleware Administering Oracle Identity Manager*.

2.3.1.7.4 Publishing a Sandbox

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

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

2.3.1.7.5 Harvesting Entitlements and Sync Catalog

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

To harvest entitlements and sync catalog:

1. Depending on the target system that you are using, run the scheduled jobs for lookup field synchronization as follows:
 - For Oracle Database:** Run the scheduled jobs listed in [Scheduled Jobs for Lookup Field Synchronization for Oracle Database](#).
 - For MSSQL:** Run the scheduled jobs listed in [Scheduled Jobs for Lookup Field Synchronization for MSSQL](#).
 - For MySQL:** Run the scheduled jobs listed in [Scheduled Job for Lookup Field Synchronization for MySQL](#).
 - For DB2:** Run the scheduled jobs listed in [Scheduled Jobs for Lookup Field Synchronization for DB2](#).
 - For Sybase:** Run the scheduled jobs listed in [Scheduled Jobs for Lookup Field Synchronization for Sybase](#).
 - For a JDBC-Based Database:** Run the scheduled jobs listed in [Configuring Scheduled Jobs for Lookup Field Synchronization](#).
2. Run the Entitlement List scheduled job to populate Entitlement Assignment schema from child process form table. See Predefined Scheduled Tasks in *Oracle Fusion Middleware Administering Oracle Identity Manager* for more information about this scheduled job.
3. Run the Catalog Synchronization Job scheduled job. See Predefined Scheduled Tasks in *Oracle Fusion Middleware Administering Oracle Identity Manager* for more information about this scheduled job.

2.3.1.7.6 Updating an Existing Application Instance with a New Form

For any changes you do in the Form Designer, you must create a new UI form and update the changes in an application instance.

To update an existing application instance with a new form:

1. Create and activate a sandbox. See *Creating a Sandbox and Activating and Deactivating a Sandbox* in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*
2. Create a new UI form for the resource. See *Creating Forms By Using the Form Designer* in *Oracle Fusion Middleware Administering Oracle Identity Manager*
3. Open the existing application instance.
4. In the Form field, select the new UI form that you created.
5. Save the application instance.
6. Publish the sandbox. See *Publishing a Sandbox* in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*

2.3.1.8 Localizing Field Labels in UI Forms for the Connector



Note:

Perform the procedure described in this section only if you are using Oracle Identity Manager release 11.1.2.x or later and you want to localize UI form field labels.

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

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

- a. Search for the following text:

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

- b. Replace with the following text:

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

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

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

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

```
<trans-unit id="$
{adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceBundl
e']}
['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.userEO.
UD_DB_ORA_U_USERNAME__c_description']}>
<source>Username</source>
</target>
</trans-unit>
<trans-unit
id="sessiondef.oracle.iam.ui.runtime.form.model.OracleDBForm.entity.Oracle
```

```
eDBForm.UD_DB_ORA_U_USERNAME__c_LABEL">
<source>Username</source>
</target>
</trans-unit>
```

- d. Open the resource file from the connector package, for example DB-UM_ja.properties, and get the value of the attribute from the file, for example, global.udf.UD_DB_ORA_U_USERNAME=\u30E6\u30FC\u30B6\u30FC\u540D.
- e. Replace the original code shown in Step 6.b with the following:

```
<trans-unit id="$
{adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceBundle']
['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.userEO.UD_DB_
ORA_U_USERNAME__c_description']}">
<source>Username</source>
<target>\u30E6\u30FC\u30B6\u30FC\u540D</target>
</trans-unit>
<trans-unit
id="sessiondef.oracle.iam.ui.runtime.form.model.OracleDBForm.entity.OracleDBFor
m.UD_DB_ORA_U_USERNAME__c_LABEL">
<source>Username</source>
<target>\u30E6\u30FC\u30B6\u30FC\u540D</target>
</trans-unit>
```

- f. Repeat Steps 6.a through 6.d for all attributes of the process form.
 - g. Save the file as BizEditorBundle_LANG_CODE.xlf. In this file name, replace LANG_CODE with the code of the language to which you are localizing.
Sample file name: BizEditorBundle_ja.xlf.
7. Repackage the ZIP file and import it into MDS.

See Also:

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

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

2.3.2 Configuring the IT Resource for the Target System

Note:

- Perform the procedure described in this section if you are using a certified database listed in [Table 2-1](#).
 - For configuring IT resource for trusted source, you must create a new IT resource of the same type definition (such as **OracleDBUM** and **MSSQLDBUM**). However, the lookup configuration for trusted source is different. See the following sections for details:
 - [Lookup Definitions for Configurations for Oracle Database](#)
 - [Lookup Definitions for Configurations for MSSQL](#)
 - [Lookup Definitions for Configurations for MySQL](#)
 - [Lookup Definitions for Configurations for DB2](#)
 - [Lookup Definitions for Configurations for Sybase](#)
- The rest of the procedure is same as described in this section.

You must specify values for the parameters of the IT resource as follows:

1. If you are using Oracle Identity Manager release 11.1.1.x:
 - a. Log in to the Administrative and User Console.
 - b. On the Welcome to Oracle Identity Manager Self Service page, click **Advanced** in the upper-right corner of the page.
 - c. On the Welcome to Oracle Identity Manager Advanced Administration page, in the Configuration region, click **Manage IT Resource**.
2. If you are using Oracle Identity Manager release 11.1.2.x or later:
 - a. Log in to Oracle Identity System Administration.
 - b. Create and activate a sandbox. For detailed instructions on creating and activating a sandbox, see *Managing Sandboxes in Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.
 - c. In the left pane, under Configuration, click **IT Resource**.
3. In the IT Resource Name field on the Manage IT Resource page, enter the name of one of the following IT resources, and then click **Search**. For example:
 - For Oracle: Oracle DB
 - For MSSQL: MSSQL DB
 - For MySQL: MySQL DB
 - For DB2: DB2
 - For Sybase: Sybase DB
4. Click the edit icon for the IT resource.

5. From the list at the top of the page, select **Details and Parameters**.
6. Specify values for the parameters of the IT resource. [Table 2-4](#) describes each parameter.

Table 2-4 IT Resource Parameters

Parameter	Description
Configuration Lookup	<p>This parameter holds the name of the lookup definition that stores configuration information for connector operations.</p> <p>If you have configured your target system as a target resource, then enter one of the following values:</p> <ul style="list-style-type: none"> • For Oracle: <code>Lookup.DBUM.Oracle.Configuration</code> • For MSSQL: <code>Lookup.DBUM.MSSQL.Configuration</code> • For MySQL: <code>Lookup.DBUM.MySQL.Configuration</code> • For DB2: <code>Lookup.DBUM.DB2.Configuration</code> • For Sybase: <code>Lookup.DBUM.Sybase.Configuration</code> <p>If you have configured your target system as a trusted source, then enter one of the following values:</p> <ul style="list-style-type: none"> • For Oracle: <code>Lookup.DBUM.Oracle.Configuration.Trusted</code> • For MSSQL: <code>Lookup.DBUM.MSSQL.Configuration.Trusted</code> • For MySQL: <code>Lookup.DBUM.MySQL.Configuration.Trusted</code> • For DB2: <code>Lookup.DBUM.DB2.Configuration.Trusted</code> • For Sybase, create a new IT resource such as <code>Lookup.DBUM.Sybase.Configuration.Trusted</code>
Connector Server Name	<p>Specify the name of the connector server IT resource.</p> <p>Sample value: <code>DBUM Connector Server</code></p>
Connection Properties	Specify the connection properties for the target system database.
Database Name	<p>This parameter specifies the database name for the SQL server.</p> <p>Sample value: <code>Master</code></p>
DB Type	<p>This field identifies database type (such as Oracle and MSSQL) and its used for loading respective scripts.</p> <p>Sample value: <code>Oracle</code></p>
JDBC Driver	<p>Depending on the target system that you are using, enter one of the following values as the JDBC driver class name:</p> <ul style="list-style-type: none"> • For Oracle: <code>oracle.jdbc.driver.OracleDriver</code> • For MSSQL: <code>microsoft.sqlserver.jdbc.SQLServerDriver</code> • For MySQL: <code>com.mysql.jdbc.Driver</code> • For DB2: <code>com.ibm.db2.jcc.DB2Driver</code> • For Sybase: <code>com.sybase.jdbc4.jdbc.SybDriver</code>
JDBC URL	<p>Specify the JDBC URL for the target system database.</p> <p>Sample Value:</p> <ul style="list-style-type: none"> • For Oracle: <code>jdbc:oracle:thin:@host:port:sid</code> If you are using Oracle 12c PDB Mode, specify <code>jdbc:oracle:thin:@//host:port/serviceName</code> as the JDBC URL. • For MSSQL: <code>jdbc:sqlserver://host:port</code> • For MySQL: <code>jdbc:mysql://host:port/database</code> • For DB2: <code>jdbc:db2://server:port/databaseName</code> • For Sybase: <code>jdbc:sybase:Tds:host:port/database</code>

Table 2-4 (Cont.) IT Resource Parameters

Parameter	Description
Login Password	<p>Enter the password for the user name of the target system account to be used for connector operations.</p> <p>Note: If you are configuring the connector for Oracle Database Vault, then you must enter the password and the user name of the account that you had created in Creating the Administrator Account on Oracle Database Vault.</p>
Login User	<p>Enter the user name of the target system account to be used for connector operations.</p> <p>For Oracle: <code>sys as sysdba</code></p> <p>For MSSQL: <code>sa</code></p> <p>For MySQL: <code>root</code></p> <p>For DB2: <code>db2admin</code></p> <p>For Sybase: <code>sa</code></p> <p>Note:</p> <ul style="list-style-type: none"> • If you are configuring the connector for Oracle Database Vault, then you must enter the user name of the account that you had created in Creating the Administrator Account on Oracle Database Vault. • The MySQL user must have all the privileges on all the schemas. <p>To grant these privileges, open the MySQL workbench, click Administration and Users and Privileges. Then, click Schema Privileges, enter * (the star wild card), and select all the privileges. Save the updates.</p>

7. To save the values, click **Update**.

2.3.3 Configuring the Connector to Support Multiple Versions of the Target System

You might want to configure the connector for different versions of the target system simultaneously. For example, you can use the connector to perform provisioning operations on SQL Server 2005, SQL Server 2008, and SQL Server 2012 simultaneously. The following example illustrates this requirement:

The London, New York, and Toronto offices of Example Multinational Inc. have their own installations of the target system. The London office has SQL Server 2005 installation, while the New York office has SQL Server 2008 installation, and the Toronto office has SQL Server 2012 installation. You have to provision resources on all the installations of DBUM simultaneously.

To meet the requirement posed by such a scenario:

You can configure a different versions of the connector bundle to simultaneously provision the resources on both the versions of the target system. The connector uses a class loading mechanism, which toggles between the different versions of the installation. You only need to place the target system-specific JAR files on the computer that hosts Oracle Identity Manager. SQL Server 2005 and 2012 need `sqljdbc.jar`, and SQL Server 2008 needs `sqljdbc4.jar`. Since there are different versions of third-party libraries, you need to create different versions of connector bundle respectively.

To configure the connector to support multiple versions of the target system:

1. From the connector package, copy the bundle JAR file in a temporary directory.

Sample JAR file: `bundle/org.identityconnectors.dbum-1.0.1116.jar`

Sample temporary directory: `c:\temp`

2. Run the following command to extract the manifest file, `META-INF/MANIFEST.MF`, from the JAR file:

```
jar -xvf org.identityconnectors.dbum-1.0.1116.jar
```

 **Note:**

You can also run the WinZip or WinRAR utility to extract the contents from the JAR file.

3. Delete the bundle JAR file in the temporary directory.
4. Update the value of **ConnectorBundle-Version** in the manifest file to a new value.

For example:

```
ConnectorBundle-Version: 1.1.1118
```

5. Copy the `sqljdbc4.jar/ojdbc6.jar` (target specific) from `DBUM_HOME/web/sqljdbc4.jar` directory or from `DBUM_HOME/web/ojdbc6.jar` directory to the lib folder of the extracted bundle jar.

6. Create a new bundle JAR file that contains the updated manifest file as follows:

- a. Open the command prompt and navigate to the temporary directory:

```
c:\temp
```

- b. Run the following command:

```
jar -cvfm org.identityconnectors.dbumintfc-1.0.1118.jar META-INF/MANIFEST.MF *
```

The new connector bundle JAR name contains the new bundle version.

7. In the case of a remote connector server, copy the new bundle JAR file in the bundles directory of the remote connector server instead of posting the JAR file to the Oracle Identity Manager database. Therefore, skip Step 8.
8. Run the Oracle Identity Manager Upload JARs utility to post the JAR file created in Step 5 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

```
OIM_HOME/server/bin/UploadJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UploadJars.sh
```

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

 **See Also:**

Migrating JARs and Resource Bundle in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for detailed information about the Upload JARs utility

9. Create a copy of the configuration lookup, for example, Lookup.DBUM.Oracle.UM.Configuration or Lookup.DBUM.MSSQL.UM.Configuration.
Ensure you update the new lookup with the bundle version.
10. Create a new DBUM IT resource definition for the new bundle. Map the **Configuration Lookup** parameter of the new IT resource to the user configuration lookup, such as Lookup.DBUM.Oracle.UM.Configuration and Lookup.DBUM.MSSQL.UM.Configuration.
The new IT resource will use the new bundle and the corresponding third-party libraries without affecting the previous installations.
11. Repeat the preceding procedure for the other versions of the target system, SQL Server 2008 and SQL Server 2012.

2.3.4 Configuring the IT Resource for the Connector Server

 **Note:**

This procedure is optional and is required only when the Connector Server is being used.

To configure or modify the IT resource for the Connector Server:

1. If you are using Oracle Identity Manager release 11.1.1.x:
 - a. Log in to the Oracle Identity Manager Administrative and User Console.
 - b. On the Welcome to Oracle Identity Manager Self Service page, click **Advanced** in the upper-right corner of the page.
 - c. On the Welcome to Oracle Identity Manager Advanced Administration page, in the Configuration region, click **Manage IT Resource**.
2. If you are using Oracle Identity Manager release 1.1.2.x or later:
 - a. Log in to Oracle Identity System Administration.
 - b. Create and activate a sandbox. For detailed instructions on creating and activating a sandbox, see *Managing Sandboxes in Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.
 - c. In the left pane, under Configuration, click **IT Resource**.

- In the IT Resource Name field on the Manage IT Resource page, enter `DBUM Connector Server` and then click **Search**. [Figure 2-1](#) shows the Manage IT Resource page.

Figure 2-1 Manage IT Resource Page for Connector Server IT Resource

Manage IT Resource

Select an IT resource and the action that you want to perform on it.

IT Resource Name

IT Resource Type

Results 1-1 of 1 First | Previous | Next | Last

IT Resource Name	IT Resource Type	Edit	Delete
DBUM Connector Server	Connector Server		

First | Previous | Next | Last

- Click the edit icon corresponding to the Connector Server IT resource.
- From the list at the top of the page, select **Details and Parameters**.
- Specify values for the parameters of the Connector Server IT resource. [Figure 2-2](#) shows the Edit IT Resource Details and Parameters page.

Figure 2-2 Edit IT Resource Details and Parameters Page for the Connector Server IT Resource

Edit IT Resource Details and Parameters

You can view additional information about this IT resource :

IT Resource Name

IT Resource Type

Remote Manager

Parameter	Value
Host	10.159.218.236
Key	*****
Port	8763
Timeout	
UseSSL	false

[Back to Search Results](#)

[Table 2-5](#) provides information about the parameters of the IT resource.

Table 2-5 Parameters of the IT Resource for the DBUM Connector Server

Parameter	Description
Host	Enter the host name or IP address of the computer hosting the Connector Server. Sample value: <code>HostName</code>
Key	Enter the key for the Connector Server.
Port	Enter the number of the port at which the Connector Server is listening. Default value: <code>8763</code>
Timeout	Enter an integer value which specifies the number of milliseconds after which the connection between the Connector Server and Oracle Identity Manager times out. If the value is zero or if no value is specified, the timeout is unlimited. Sample value: <code>0</code> (recommended value)
UseSSL	Enter <code>true</code> to specify that you will configure SSL between Oracle Identity Manager and the Connector Server. Otherwise, enter <code>false</code> . Default value: <code>false</code> See Also: Configuring Secure Communication Between Oracle Database and Oracle Identity Manager and Configuring Secure Communication Between MSSQL and Oracle Identity Manager for information about enabling SSL.

- To save the values, click **Update**.

2.3.5 Enabling Logging for Connector Server

When you enable logging, the connector server stores in a log file information about events that occur during the course of provisioning and reconciliation operations for different statuses. By default, the connector server logs are set at INFO level and you can change this level to the following:

- **Error**
This level enables logging of information about errors that might allow connector server to continue running.
- **WARNING**
This level enables logging of information about potentially harmful situations.
- **INFO**
This level enables logging of messages that highlight the progress of the operation.
- **FINE, FINER, FINEST**
These levels enable logging of information about fine-grained events, where FINEST logs information about all events.

To enable the logging information for the connector server:

- Go to `CONNECTOR_SERVER_HOME/Conf`.
- Open the `logging.properties` file.
- Edit the following entry:
`.level=INFO`
- Save and close the file.

5. Restart the connector server.

2.4 Upgrading the Connector

If you have already deployed an earlier release of this connector, then upgrade the connector to the current release 11.1.1.6.0.

Note:

Before you perform the upgrade procedure:

- It is strongly recommended that you create a backup of the Oracle Identity Manager database. Refer to the database documentation for information about creating a backup.
- As a best practice, perform the upgrade procedure in a test environment initially.

See Also:

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

The following sections discuss the procedure to upgrade the connector:

- [Preupgrade Steps](#)
- [Upgrade Steps](#)
- [Postupgrade Steps](#)

2.4.1 Preupgrade Steps

Perform the following preupgrade steps:

1. Perform a reconciliation run to fetch all latest updates to Oracle Identity Manager.
2. Define the source connector (an earlier release of the connector that must be upgraded) in Oracle Identity Manager. You define the source connector to update the Deployment Manager XML file with all customization changes made to the connector.
3. If required, create the connector XML file for a clone of the source connector.
4. Disable all the scheduled jobs.

2.4.2 Upgrade Steps

The following is the summary of the procedure to upgrade the connector:

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

Perform the upgrade procedure by using the wizard mode.

 **Note:**

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

- Production Environment

Perform the upgrade procedure by using the silent mode.

2.4.3 Postupgrade Steps

Perform the following procedure:

1. Upload new connector jars as:
 - a. Use `$ORACLE_HOME/bin/UploadJars.sh` utility for uploading connector jars.
 - b. Upload `bundle/org.identityconnectors.dbum-1.0.1116.jar` as ICFBundle.

 **Note:**

If you have to add a third-party JAR:

- Navigate to the bundle directory.
- Create `/lib` folder and drop the third party jar in that folder.
- Update the bundle with library `"jar uvf org.identityconnectors.dbum-1.0.1116.jar lib/FILE_NAME"`.

- c. Upload `lib/DBUM-oim-integration.jar` as JavaTask.
2. Run the Form Version Control (FVC) utility to manage data changes on a form after an upgrade operation. To do so:
 - a. In a text editor, open the `fvc.properties` file located in the `OIM_DC_HOME` directory and include the following entries:

For Oracle User:

```
ResourceObject;Oracle DB User
FormName;UD_DB_ORA_U
FromVersion;Version 0
ToVersion;v_22
ParentParent;UD_DB_ORA_U_USERNAME;UD_DB_ORA_U_RET_ID
```

For MSSQL UserLogin:

```
ResourceObject;MSSQL DB User Login
FormName;UD_DB_SQL_L
FromVersion;Version 0
ToVersion;v_11.1.1.1.8.0
ParentParent;UD_DB_SQL_L_LOGIN;UD_DB_SQL_L_REFID
```

For MSSQL User:

```
ResourceObject;MSSQL DB User
FormName;UD_DB_SQL_U
FromVersion;Version 7
ToVersion;v_11.1.1.1.8.0
ParentParent;UD_DB_SQL_U_USERNAME;UD_DB_SQL_U_REFID
```

For MySQL User:

```
ResourceObject;MySQL DB User
FormName;UD_DB_MYS_U
FromVersion;6
ToVersion;1
ParentParent;UD_DB_MYS_U_USER_NAME;UD_DB_MYS_U_RET_ID
```

For DB2 User:

```
ResourceObject;DB2 DB User
FormName;UD_DB_DB2_U
FromVersion;8
ToVersion;10
ParentParent;UD_DB_DB2_U_USERNAME;UD_DB_DB2_U_RET_ID
```

For Sybase User:

```
ResourceObject;Sybase DB User
FormName;UD_DB_SYB_U
FromVersion;8
ToVersion;10
ParentParent;UD_DB_SYB_U_USERNAME;UD_DB_SYB_U_RETURN_ID
```

For Sybase UserLogin:

```
ResourceObject;Sybase DB User Login
FormName;UD_DB_SYB_L
FromVersion;5
ToVersion;10
ParentParent;UD_DB_SYB_L_LOGIN;UD_DB_SYB_L_RET_ID
```

- b. Run the FVC utility. This utility is copied into the following directory when you install the design console:

For Microsoft Windows:

```
OIM_DC_HOME/fvcutil.bat
```

For UNIX:

```
OIM_DC_HOME/fvcutil.sh
```

When you run this utility, you are prompted to enter the login credentials of the Oracle Identity Manager administrator, and the logger level and log file location.

3. Run the PostUpgradeScript.sql script as follows:
 - a. Connect to the Oracle Identity Manager database by using the OIM User credentials.
 - b. Run the PostUpgradeScript:
 - For Oracle: PostUpgradeScriptOracleDBUM.sql
 - For MSSQL: PostUpgradeScriptMSSQLDBUM.sql
 - For MSSQL: PostUpgradeScriptMySQLDBUM.sql
 - For MSSQL: PostUpgradeScriptDB2DBUM.sql
 - For MSSQL: PostUpgradeScriptSybaseDBUM.sql

This script will upgrade the IT resource only for the required database. For example, after upgrading the connector, you can upgrade the IT resource only for the MSSQL database by running `PostUpgradeScriptMSSQLDBUM.sql`. The script will not upgrade the IT resource for the Oracle database.

4. If you are using Oracle Identity Manager release 11.1.2.x or later, then all changes made to the Form Designer of the Design Console must be done in a new UI form as follows:
 - a. Log in to Oracle Identity System Administration.
 - b. Create and activate a sandbox. For more information, see step 2 of [Configuring Oracle Identity Manager Release 11.1.2 or Later](#).
 - c. Create a new UI form to view the upgraded fields. See step 3 of [Configuring Oracle Identity Manager Release 11.1.2 or Later](#) for more information about creating a UI form.
 - d. Associate the newly created UI form with the application instance of your target system. To do so, open the existing application instance for your resource, from the Form field, select the form (created in step 2.c) and then save the application instance.
 - e. Publish the sandbox. See step 5 of [Configuring Oracle Identity Manager Release 11.1.2 or Later](#) for more information.
5. Configure the upgraded IT resource of the source connector. See [Configuring the IT Resource for the Target System](#) for information about configuring the IT resource.
6. Deploy the Connector Server. See [Installing and Configuring the Connector Server](#) and [Installing the Connector on the Connector Server](#) for more information.
7. Configure the latest token value of the scheduled job as follows:

The following scheduled jobs contain the Latest Token attribute:

For Oracle

- DBUM Oracle User Target Reconciliation
- DBUM Oracle User Trusted Reconciliation

For MSSQL:

- DBUM MSSQL Trusted Reconciliation
- DBUM MSSQL User Login Target Reconciliation
- DBUM MSSQL User Target Reconciliation

After upgrading the connector, you can perform either full reconciliation or incremental reconciliation. This ensures that records created or modified since the last reconciliation run (the one that you performed in [Preupgrade Steps](#)) are fetched into Oracle Identity Manager. From the next reconciliation run onward, the reconciliation engine automatically enters a value for the Latest Token attribute.

See [Reconciliation from Oracle Database](#) and [Reconciliation from MSSQL](#) for more information about performing full or incremental reconciliation for Oracle and MSSQL databases respectively.

 **Note:**

If there are customizations in the query files, to include custom parameters, and for transformation/validation of data during reconciliation/ provisioning, then the same customizations have to be performed in the respective query files after upgrading the connector.

2.5 Postcloning Steps

You can clone the connector by setting new names for some of the objects that comprise the connector. The outcome of the process is a new connector XML file. Most of the connector objects, such as Resource Object, Process Definition, Process Form, IT Resource Type Definition, IT Resource Instances, Lookup Definitions, Adapters, Reconciliation Rules and so on in the new connector XML file have new names.

 **See Also:**

Cloning Connectors in *Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed information about cloning connectors and the steps mentioned in this section

This section contains the following topics:

- [Connector Objects](#)
- [Postcloning Configuration for User Accounts](#)

2.5.1 Connector Objects

After a copy of the connector is created by setting new names for connector objects, some objects might contain the details of the old connector objects. Therefore, you must modify the following Oracle Identity Manager objects to replace the base connector artifacts or attribute references with the corresponding cloned artifacts or attributes:

- **IT Resource**
The cloned connector has its own set of IT resources. You must configure both the cloned connector IT resources and Connector Server IT resources, and provide the reference of the cloned Connector Server IT Resource in the cloned connector IT resource. Ensure you use the configuration lookup definition of the cloned connector.
- **Scheduled Job**
The values of the Resource Object Name and IT Resource scheduled job attributes in the cloned connector refer to the values of the base connector. Therefore, these values (values of the Resource Object Name and IT resource scheduled job attributes that refer to the base connector) must be replaced with the new cloned connector artifacts.
- **Lookup Definition**
No change is required to be made in any of the cloned lookup definitions. All cloned lookup definitions contain proper lookup entries.

- Process Tasks

After cloning, you notice that all event handlers attached to the process tasks are the cloned ones. Therefore, no changes are required for process tasks in parent forms. This is because the adapter mappings for all process tasks related to parent forms are updated with cloned artifacts.

- Localization Properties

You must update the resource bundle of a user locale with new names of the process form attributes for proper translations after cloning the connector. You can modify the properties file of your locale in the resources directory of the connector bundle.

For example, the process form (UD_DB_SQL_U) attributes are referenced in the Japanese properties file, DB-UM_ja.properties, as `global.udf.UD_DB_SQL_U_USERNAME`. During cloning, if you change the process form name from `UD_DB_SQLCLONED_U` to `global.udf.UD_DB_SQLCLONED_U_USERNAME`, then you must add the process form attributes to `global.udf.UD_DB_SQL_U_USERNAME`.

2.5.2 Postcloning Configuration for User Accounts

You must perform the postcloning steps to change the literal values.

This configuration change is related to a child form. When you add a role, privilege, tablespace or schema to an account, perform the following steps:

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Process Management**, and then double-click **Process Definition**.
3. Depending on the target system being used, search for and open one of the following cloned process definitions:
 - For Oracle: Oracle DB User
 - For MSSQL: MSSQL DB User
 - For MySQL: My SQL DB User
 - For DB2: DB2 DB User
 - For Sybase: Sybase DB User Login
4. If you are using Oracle Database as the target system, then double-click the **Add Role or Grant Process** task.

The Editing Task: Add Role or Grant Process dialog box is displayed.
5. On the **Integration tab**, in the table in the **Adapter Variables** region, click the **processInstanceKey** variable, and then click **Map**.
6. In the **Edit Data Mapping For Variable** dialog box, create the following mapping:
 - Variable Name: processInstanceKey
 - Data Type: Long
 - Map To: Response Code
 - Qualifier: Process Instance
7. Click the Save icon and close the dialog box.

8. Perform Steps 5 through 7 for the remaining variables listed in the **Adapter Variables** region. [Table 2-6](#) lists values that you must select from the Data Type, Map To, Qualifier, and Literal Value lists for each variable.

Table 2-6 Mappings for DBUM Oracle Event Handler/Adapter

Variable Name	Data Type	Map to	Qualifier	Literal Value
Adapter return value	Object	Response Code	NA	NA
objectType	String	Literal	String	User
itResourceName [Cloned IT Resource Name]	String	Literal	String	UD_DB_ORA_U_ITRES 1
childTableName[Clon ed Child Form]	String	Literal	String	UD_DB_ORA_R1

 **Note:**

Open the childTableName mapping and change the Literal value to the new value (the cloned value).

9. Click the Save icon on the Process Definition form.
10. Repeat Steps 4 through 9 to update **Revoke Role** and **Role Updated** process tasks.
11. If you are using Oracle Database or My SQL as the target system, repeat Steps 4 through 9 for the **Add Privilege or Grant, Update Privilege**, and **Revoke Privilege** process tasks.
12. If you are using Sybase or MSSQL as the target system, repeat Steps 4 through 9 for the **Add Role, Role Updated**, and **Revoke Role** process tasks, which only supports roles.
13. If you are using DB2 as the target system, repeat Steps 4 through 9 for the **Add Tablespace, Tablespace Updated, Delete Tablespace, Add Schema, Schema Updated** and **Delete Schema** process tasks, which supports tablespaces and schemas.

 **Note:**

After post cloning steps, user must create a new application instance and UI Form for the cloned DBUM connector as it is done for any DBUM connector setup.

3

Using and Extending the Connector for MSSQL

You can use the Database User Management Connector for MSSQL for performing reconciliation and provisioning operations after configuring it to meet your requirements. You can also extend the functionality of the connector to address your specific business requirements.

This chapter contains the following topics:

Note:

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

For Oracle Identity Manager hosted on a Microsoft Windows computer, if you have a previously installed connector, then you must extract the connector bundle zip file again before installing a new connector.

- [Configuring Secure Communication Between MSSQL and Oracle Identity Manager](#)
- [Guidelines on Determining Values for the JDBC URL and Connection Properties Parameters for MSSQL](#)
- [Lookup Definitions for MSSQL](#)
- [Scheduled Jobs for MSSQL](#)
- [Reconciliation from MSSQL](#)
- [Provisioning for MSSQL](#)
- [Extending the Connector for MSSQL](#)

3.1 Configuring Secure Communication Between MSSQL and Oracle Identity Manager

Note:

It is recommended that you perform the procedure described in this section to secure communication between the target system and Oracle Identity Manager.

Perform the following steps to configure secure communication between Microsoft SQL Server and Oracle Identity Manager:

1. See Microsoft SQL Server documentation for information about enabling SSL communication between Microsoft SQL Server and a client system. In this context, the client is Oracle Identity Manager.

Export the certificate on the Microsoft SQL Server host computer, and then restart the database service.

2. Copy the certificate to the Oracle Identity Manager host computer.
3. Import the certificate into the JVM truststore of the application server on which Oracle Identity Manager is running.

To import the certificate into the truststore, run the following command:

```
..\..\bin\keytool -import -file FILE_LOCATION -keystore TRUSTSTORE_LOCATION -storepass TRUSTSTORE_PASSWORD -trustcacerts -alias ALIAS
```

In this command:

- Replace *FILE_LOCATION* with the full path and name of the certificate file.
- Replace *ALIAS* with an alias for the certificate.
- Replace *TRUSTSTORE_PASSWORD* with a password for the truststore.
- Replace *TRUSTSTORE_LOCATION* with one of the truststore paths from [Table 3-1](#). This table shows the location of the truststore for each of the supported application servers.

 **Note:**

In an Oracle Identity Manager cluster, import the file into the truststore on each node of the cluster.

Table 3-1 Truststore Locations on Supported Application Servers

Application Server	Truststore Location
Oracle WebLogic Server	<ul style="list-style-type: none"> • If you are using Oracle jrockit_R27.3.1-jdk, then import the certificate into the keystore in the following directory: <i>JROCKIT_HOME</i>/jre/lib/security • If you are using the default Oracle WebLogic Server JDK, then import the certificate into the keystore in following directory: <i>WEBLOGIC_HOME</i>/java/jre/lib/security/cacerts • If you are using a JDK other than Oracle jrockit_R27.3.1-jdk or Oracle WebLogic Server JDK, then import the certificate into your keystore at the following directory: <i>JAVA_HOME</i>/jre/lib/security/cacerts

4. To enable secure communication between Microsoft SQL Server and Oracle Identity Manager, set the value of the UseSSL IT resource parameter to `true`. You must provide a value for this parameter while performing the procedure described in [Configuring the IT Resource for the Connector Server](#).

3.2 Guidelines on Determining Values for the JDBC URL and Connection Properties Parameters for MSSQL

This section discusses the JDBC URL and Connection Properties parameters. You apply the information in this section while performing the procedure described in [Configuring the IT Resource for the Target System](#).

The values that you specify for the Database URL and Connection Properties parameters depend on the target system:



Note:

In Microsoft SQL Server documentation, the term "connection URL" is used instead of "JDBC URL."

- **JDBC URL parameter**

Enter the following component of the connection URL as the value of the JDBC URL provider:

```
jdbc:sqlserver://[SERVER_NAME][:PORT_NUMBER];database=DATABASE_NAME]
```

In this format:

- *SERVER_NAME* is the IP address (not the host name) of the target system host computer.
- *PORT_NUMBER* is the port at which the target system database is listening.
- *DATABASE_NAME* is the name of the database we are connecting.



Note:

If you specify database names both in the Database Name parameter and in the JDBC URL parameter, then the connector connects to the database specified in the Database Name parameter.

The following is a sample value for the Database URL parameter:

```
jdbc:sqlserver://192.168.16.76:1433;database=model
```

- **Connection Properties parameter**

Enter the following component of the connection URL as the value of the Connection Properties parameter:

```
[;PROPERTY=VALUE[;PROPERTY=VALUE]] . . .
```

In this format:

- *PROPERTY* is the name of one or more database connection properties, such as `applicationName` and `disableStatementPooling`.

- *VALUE* is the value of each database connection property whose name you specify by using the *PROPERTY* placeholder.

 **Note:**

Semicolons must be changed to number signs (#) in the value that you specify.

The following is a sample value for the Connection Properties parameter:

```
databaseName=sales#port=1433
```

If you enable SSL communication between Microsoft SQL Server and Oracle Identity Manager, then you must include the `encrypt` and `hostNameInCertificate` properties in the value that you specify for the Connection Properties parameter. In other words, the following must be part of the string that you enter as the value of the parameter:

```
encrypt=true#hostNameInCertificate=HOST_NAME
```

Replace *HOST_NAME* with the host name given in the certificate that you use.

In addition, you must specify the location of the truststore if you import the certificate into a truststore other than the JVM truststore of Oracle Identity Manager. To specify the location of the truststore, include the following properties in the value that you specify for the Connection Properties parameter:

```
encrypt=true#hostNameInCertificate=HOST_NAME#trustStore=STORE_LOCATION#trustStorePassword=STORE_PASSWORD
```

When you specify this value, replace *STORE_LOCATION* with the full path and name of the truststore, and replace *STORE_PASSWORD* with the password of the truststore.

3.3 Lookup Definitions for MSSQL

This section contains the following information:

- [Lookup Definitions Synchronized with MSSQL](#)
- [Lookup Definitions for Configurations for MSSQL](#)
- [Lookup Definitions for Attribute Mappings for MSSQL Login Entity](#)
- [Lookup Definitions for Attribute Mappings for MSSQL User Entity](#)
- [Lookup Definitions for Exclusion Lists for MSSQL](#)
- [Lookup Definitions for Transformation of Data in MSSQL](#)
- [Lookup Definition for Validation of Data in MSSQL](#)

 **Tip:**

You must provide Decode values for some of the entries of the lookup definitions listed in this topic. To set a Decode value for an entry in a lookup definition:

1. On the Design Console, expand **Administration**, and then double-click **Lookup Definition**.
2. Search for and open the lookup definition that you want to modify.
3. Enter the value in the **Decode** column for the Code Key that you want to set.
4. Click the save icon.

3.3.1 Lookup Definitions Synchronized with MSSQL

During a provisioning operation, you use a lookup field on the process form to specify a single value from a set of values. For example, you use the Role lookup field to select a role to be assigned to the user from the list of available roles. When you deploy the connector, lookup definitions corresponding to the lookup fields on the target system are created in Oracle Identity Manager. Lookup field synchronization involves copying additions or changes made to the target system lookup fields into the lookup definitions in Oracle Identity Manager.

The connector provides predefined SQL queries for fetching values from the target system lookup fields into the lookup definitions in Oracle Identity Manager. These predefined SQL queries are stored in the `LoVSearch.queries` file within the connector bundle.

After lookup definition synchronization, data is stored in the following format:

- Code Key value: `IT_RESOURCE_KEY~LOOKUP_FIELD_ID`

In this format:

- `IT_RESOURCE_KEY` is the numeric code assigned to each IT resource in Oracle Identity Manager.
- `LOOKUP_FIELD_ID` is the target system code assigned to each lookup field entry.

Sample value: `1~SYS_ADM`

- Decode value: `IT_RESOURCE_NAME~LOOKUP_FIELD_ID`

In this format:

- `IT_RESOURCE_NAME` is the name of the IT resource in Oracle Identity Manager.
- `LOOKUP_FIELD_ID` is the target system code assigned to each lookup field entry.

While performing a provisioning operation in Oracle Identity Self Service, you select the IT resource for the target system on which you want to perform the operation. When you perform this action, the lookup definitions on the page are automatically populated with values corresponding to the IT resource (target system installation) that you select. If your environment has multiple installations of the target system, then values corresponding to all IT resources are displayed.

[Table 3-2](#) lists column names of the tables in Microsoft SQL Server that are synchronized with their corresponding lookup definitions in Oracle Identity Manager.

Table 3-2 Lookup Definitions Synchronized with Microsoft SQL Server

Lookup Definition	Target Column Name
Lookup.DBUM.MSSQL.AuthType	authType
Lookup.DBUM.MSSQL.DBNames	name
Lookup.DBUM.MSSQL.DBRoles	name
Lookup.DBUM.MSSQL.DefaultLang	alias

In Microsoft SQL server, you can create an account (login or user) that uses either Windows authentication or SQL server authentication.

The Lookup.DBUM.MSSQL.AuthType lookup definition holds information about authentication types that you can select for a target system account (login or user) that you create through Oracle Identity Manager.

MSSQL can operate in one of two authentication types:

- **Windows Authentication**
Windows Authentication allows a user to connect through a Microsoft Windows user account. This is the default type of authentication.
- **SQL Server Authentication**
SQL Server authentication relies on the internal user list maintained by the SQL Server computer. This list does not include Windows users, and is specific to the SQL Server computer. Users are created and configured using the SQL Server Enterprise Manager.

The following is the format of the Code Key and Decode values in this lookup definition:

- **Code Key:** Type of authentication
- **Decode:** Description of the type of authentication

[Table 3-3](#) lists the default entries in this lookup definition.

Table 3-3 Entries in the Lookup.DBUM.MSSQL.AuthType Lookup Definition

Code Key	Decode
SQL_SERVER_AUTHENTICATION	SQL_SERVER_AUTHENTICATION
WINDOWS_AUTHENTICATION	WINDOWS_AUTHENTICATION

3.3.2 Lookup Definitions for Configurations for MSSQL

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

This section provides information about the following lookup definitions

- [Lookup.DBUM.MSSQL.Configuration](#)

- [Lookup.DBUM.MSSQL.Login.Configuration](#)
- [Lookup.DBUM.MSSQL.UM.Configuration](#)
- [Lookup.DBUM.MSSQL.Configuration.Trusted](#)
- [Lookup.DBUM.MSSQL.UM.Configuration.Trusted](#)

3.3.2.1 Lookup.DBUM.MSSQL.Configuration

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



Note:

You cannot add entries to this lookup definition. However, you can modify the Decode values of the existing entries.

Table 3-4 lists the default entries in this lookup definition.

Table 3-4 Entries in the Lookup.DBUM.MSSQL.Configuration Lookup Definition

Code Key	Decode	Description
Bundle Name	org.identityconnectors.dbum	This entry holds the name of the connector bundle package. Do not modify this entry.
Bundle Version	1.0.1116	This entry holds the version of the connector bundle class. Do not modify this entry.
Connector Name	org.identityconnectors.dbum.DBUMConnector	This entry holds the name of the connector class. Do not modify this entry.
disableValuesSet	"true"	This entry is used internally. Do not modify this entry.
User Configuration Lookup	Lookup.DBUM.MSSQL.UM.Configuration	This entry holds the name of the lookup definition that contains user-specific configuration properties. Do not modify this entry.
USERLOGIN Configuration Lookup	Lookup.DBUM.MSSQL.Login.Configuration	This entry holds the name of the lookup definition that contains login-specific configuration properties. Do not modify this entry.

3.3.2.2 Lookup.DBUM.MSSQL.Login.Configuration

The Lookup.DBUM.MSSQL.Login.Configuration lookup definition holds configuration entries for Microsoft SQL Server login entity that are used during target resource reconciliation and provisioning operations.

Table 3-5 lists the default entries in this lookup definition.

Table 3-5 Entries in Lookup.DBUM.MSSQL.Login.Configuration

Code Key	Decode Key
Provisioning Attribute Map	Lookup.DBUM.MSSQL.Login.ProvAttrMap
Provisioning Exclusion List	Lookup.DBUM.MSSQL.Login.ProvExclusions

Table 3-5 (Cont.) Entries in Lookup.DBUM.MSSQL.Login.Configuration

Code Key	Decode Key
Provisioning Validation Lookup	Lookup.DBUM.MSSQL.Login.ProvValidations
Recon Attribute Defaults	Lookup.DBUM.MSSQL.Login.ReconDefaults
Recon Attribute Map	Lookup.DBUM.MSSQL.Login.ReconAttrMap
Recon Exclusion List	Lookup.DBUM.MSSQL.Login.ReconExclusions
Recon Transformation Lookup	Lookup.DBUM.MSSQL.Login.ReconTransformation
Recon Validation Lookup	Lookup.DBUM.MSSQL.Login.ReconValidation

3.3.2.3 Lookup.DBUM.MSSQL.UM.Configuration

The Lookup.DBUM.MSSQL.UM.Configuration lookup definition holds connector configuration entries for Microsoft SQL Server user entity that are used during target resource reconciliation and provisioning operations.

Table 3-6 lists the default entries in this lookup definition.

Table 3-6 Entries in Lookup.DBUM.MSSQL.UM.Configuration

Code Key	Dcode Key
Provisioning Attribute Map	Lookup.DBUM.MSSQL.UM.ProvAttrMap
Provisioning Exclusion List	Lookup.DBUM.MSSQL.UM.ProvExclusions
Provisioning Validation Lookup	Lookup.DBUM.MSSQL.UM.ProvValidations
Recon Attribute Defaults	Lookup.DBUM.MSSQL.UM.ReconDefaults
Recon Attribute Map	Lookup.DBUM.MSSQL.UM.ReconAttrMap
Recon Exclusion List	Lookup.DBUM.MSSQL.UM.ReconExclusions
Recon Transformation Lookup	Lookup.DBUM.MSSQL.UM.ReconTransformations
Recon Validation Lookup	Lookup.DBUM.MSSQL.UM.ReconValidation

3.3.2.4 Lookup.DBUM.MSSQL.Configuration.Trusted

The Lookup.DBUM.MSSQL.Configuration.Trusted lookup definition holds connector configuration entries that are used during reconciliation and provisioning operations in trusted source mode.

 **Note:**

You cannot add entries to this lookup definition. However, you can modify the Decode values of the existing entries.

Table 3-7 lists the default entries in this lookup definition.

Table 3-7 Entries in the Lookup.DBUM.MSSQL.Configuration.Trusted Lookup Definition

Code Key	Decode	Description
Bundle Name	org.identityconnectors.dbum	This entry holds the name of the connector bundle package. Do not modify this entry.
Bundle Version	1.0.1116	This entry holds the version of the connector bundle class. Do not modify this entry.
Connector Name	org.identityconnectors.dbum.DBUMConnector	This entry holds the name of the connector class. Do not modify this entry.
disableValuesSet	"true"	This entry is used internally. Do not modify this entry.
USERLOGIN Configuration Lookup	Lookup.DBUM.MSSQL.UM .Configuration.Trusted	This entry holds the name of the lookup definition that contains login-specific configuration properties. Do not modify this entry.

3.3.2.5 Lookup.DBUM.MSSQL.UM.Configuration.Trusted

The Lookup.DBUM.MSSQL.UM.Configuration.Trusted lookup definition holds connector configuration entries for Microsoft SQL Server user entity that are used during reconciliation and provisioning operations in trusted source mode.

[Table 3-8](#) lists the default entries in this lookup definition.

Table 3-8 Entries in Lookup.DBUM.MSSQL.UM.Configuration.Trusted

Code Key	Dcode Key
Recon Attribute Defaults	Lookup.DBUM.MSSQL.UM.ReconDefaults.Trusted
Recon Attribute Map	Lookup.DBUM.MSSQL.UM.ReconAttrMap.Trusted
Recon Exclusion List	Lookup.DBUM.MSSQL.UM.ExclusionList.Trusted
Recon Transformation Lookup	Lookup.DBUM.MSSQL.UM.ReconTransformations.Trusted
Recon Validation Lookup	Lookup.DBUM.MSSQL.UM.ReconValidation.Trusted

3.3.3 Lookup Definitions for Attribute Mappings for MSSQL Login Entity

This section describes the following lookup definitions:

- [Lookup.DBUM.MSSQL.Login.ProvAttrMap](#)
- [Lookup.DBUM.MSSQL.Login.ReconAttrMap](#)
- [Lookup.DBUM.MSSQL.Login.ReconDefaults](#)

3.3.3.1 Lookup.DBUM.MSSQL.Login.ProvAttrMap

The Lookup.DBUM.MSSQL.Login.ProvAttrMap lookup definition holds mappings for Microsoft SQL Server login entity between process form fields (Code Key values) and target system attributes (Decode values) used during provisioning operations.

[Table 3-9](#) lists the default entries in this lookup definition.

Table 3-9 Entries in Lookup.DBUM.MSSQL.Login.ProvAttrMap

Code Key	Decode Key
Authentication Type	authType
Default DataBase[LOOKUP]	defaultDatabase
Default Language[LOOKUP]	defaultLanguage
Login Name	__NAME__
Password	__PASSWORD__
Reference ID	__UID__

3.3.3.2 Lookup.DBUM.MSSQL.Login.ReconAttrMap

The Lookup.DBUM.MSSQL.Login.ReconAttrMap lookup definition holds mappings for Microsoft SQL Server login entity between process form fields (Code Key values) and target system attributes (Decode values) used during reconciliation operations.

[Table 3-10](#) lists the default entries in this lookup definition.

Table 3-10 Entries in Lookup.DBUM.MSSQL.Login.ReconAttrMap

Code Key	Decode Key
Authentication Type	authType
Default Database Name[LOOKUP]	defaultDatabase
Default Language[LOOKUP]	defaultLanguage
Login Name	__UID__
Reference ID	__UID__
Status	__ENABLE__

3.3.3.3 Lookup.DBUM.MSSQL.Login.ReconDefaults

This lookup is empty by default.

3.3.4 Lookup Definitions for Attribute Mappings for MSSQL User Entity

This section describes the following lookup definitions:

- [Lookup.DBUM.MSSQL.UM.ProvAttrMap](#)
- [Lookup.DBUM.MSSQL.UM.ReconAttrMap](#)
- [Lookup.DBUM.MSSQL.UM.ReconDefaults](#)
- [Lookup.DBUM.MSSQL.UM.ReconDefaults.Trusted](#)
- [Lookup.DBUM.MSSQL.UM.ReconAttrMap.Trusted](#)

3.3.4.1 Lookup.DBUM.MSSQL.UM.ProvAttrMap

The Lookup.DBUM.MSSQL.UM.ProvAttrMap lookup definition holds mappings between process form fields (Code Key values) and target system attributes (Decode values) used during provisioning operations.

[Table 3-11](#) lists the default entries in this lookup definition.

Table 3-11 Entries in Lookup.DBUM.MSSQL.UM.ProvAttrMap

Code Key	Decode Key
Database Name[LOOKUP]	defaultDatabase
Login Name	loginName
Reference ID	__UID__
UD_DB_SQL_R~Role[LOOKUP]	roles~DBRole~__NAME__
Username	__NAME__

3.3.4.2 Lookup.DBUM.MSSQL.UM.ReconAttrMap

The Lookup.DBUM.MSSQL.UM.ReconAttrMap lookup definition holds mappings between resource object fields (Code Key values) and target system attributes (Decode values) used during reconciliation operations.

[Table 3-12](#) lists the default entries in this lookup definition.

Table 3-12 Entries in Lookup.DBUM.MSSQL.UM.ReconAttrMap

Code Key	Decode Key
Database Name	defaultDatabase
Login Name	loginName
Reference ID	__UID__
Role List~Role[LOOKUP]	roles~DBRole~__NAME__
User Name	__UID__

3.3.4.3 Lookup.DBUM.MSSQL.UM.ReconDefaults

The Lookup.DBUM.MSSQL.UM.ReconDefaults lookup definition holds the following entries:

[Table 3-13](#) lists the default entries in this lookup definition.

Table 3-13 Entries in Lookup.DBUM.MSSQL.UM.ReconDefaults

Code Key	Decode Key
Login Name	WITHOUT LOGIN

3.3.4.4 Lookup.DBUM.MSSQL.UM.ReconDefaults.Trusted

The Lookup.DBUM.MSSQL.UM.ReconDefaults.Trusted lookup definition holds the following entries:

[Table 3-14](#) lists the default entries in this lookup definition.

Table 3-14 Entries in Lookup.DBUM.MSSQL.UM.ReconDefaults.Trusted

Code Key	Decode Key
Empl Type	Full-Time
Organization Name	Xellerate Users
Status	Active
User Type	End-User

3.3.4.5 Lookup.DBUM.MSSQL.UM.ReconAttrMap.Trusted

The Lookup.DBUM.MSSQL.UM.ReconAttrMap.Trusted lookup definition holds mappings between resource object fields (Code Key values) and target system attributes (Decode values) used during reconciliation operations in trusted source mode.

[Table 3-15](#) lists the default entries in this lookup definition.

Table 3-15 Entries in Lookup.DBUM.MSSQL.UM.ReconAttrMap.Trusted

Code Key	Decode Key
First Name	__UID__
Status[TRUSTED]	__ENABLE__
User ID	__UID__

3.3.5 Lookup Definitions for Exclusion Lists for MSSQL

This section describes the lookup definitions that hold resources for which you do not want to perform provisioning and reconciliation operations.

You can use one of the following lookups for MSSQL login entity:

- For provisioning operations: Lookup.DBUM.MSSQL.Login.ProvExclusions
- For reconciliation operations: Lookup.DBUM.MSSQL.Login.ReconExclusions

You can use one of the following lookups for MSSQL user entity:

- For provisioning operations: Lookup.DBUM.MSSQL.UM.ProvExclusions
- For reconciliation operations: Lookup.DBUM.MSSQL.UM.ReconExclusions
- For target system configured as trusted source:
Lookup.DBUM.MSSQL.UM.ExclusionList.Trusted

The following is the format of the values stored in these lookups:

Code Key	Decode	Sample Values
Login Name	User ID of a user	Code Key: Login Name Decode: User001
Login Name with the [PATTERN] suffix	A regular expression supported by the representation in the <code>java.util.regex.Pattern</code> class	Code Key: Login Name[PATTERN] To exclude users matching any of the user ID 's User001, User002, User088, then: Decode: User001 User002 User088 To exclude users whose user ID 's start with 00012, then: Decode: 00012* See Also: For information about the supported patterns, visit http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html

[Configuring Resource Exclusion Lists for MSSQL](#) describes the procedure to add entries in these lookup definitions.

3.3.6 Lookup Definitions for Transformation of Data in MSSQL

Depending on how the target system is configured, you can use one of the following lookups to enable transformation of data during reconciliation operations:

- For Microsoft SQL Server login entity:
Lookup.DBUM.MSSQL.Login.ReconTransformation

Table 3-16 Entries in Lookup.DBUM.MSSQL.Login.ReconTransformation

Code Key	Decode Key
Authentication Type	mssql.oimcp.dbum.transformations.AuthTypeTransformationImpl

- For Microsoft SQL Server user entity: Lookup.DBUM.MSSQL.UM.ReconTransformations
- For Microsoft SQL Server user entity in trusted source mode:
Lookup.DBUM.MSSQL.UM.ReconTransformations.Trusted

[Configuring Transformation of Data During User Reconciliation for MSSQL](#) describes the procedure to add entries in these lookup definitions.

3.3.7 Lookup Definition for Validation of Data in MSSQL

Depending on how the target system is configured, you can use one of the following lookups to enable validation of data:

- For Microsoft SQL Server login entity during provisioning:
Lookup.DBUM.MSSQL.Login.ProvValidations
- For Microsoft SQL Server login entity during reconciliation:
Lookup.DBUM.MSSQL.Login.ReconValidation
- For Microsoft SQL Server user entity during provisioning:
Lookup.DBUM.MSSQL.UM.ProvValidations

- For Microsoft SQL Server user entity during reconciliation:
Lookup.DBUM.MSSQL.UM.ReconValidation

[Configuring Validation of Data During Reconciliation and Provisioning for MSSQL](#) describes the procedure to add entries in this lookup definition.

3.4 Scheduled Jobs for MSSQL

When you run the Connector Installer or import the connector XML file, the scheduled jobs are automatically created in Oracle Identity Manager.

This section describes the following topics:

- [Scheduled Jobs for Lookup Field Synchronization for MSSQL](#)
- [Attributes of the Scheduled Jobs for Reconciliation of MSSQL User and Login Entities](#)
- [Attributes of the Scheduled Jobs for MSSQL Delete Operations](#)
- [Configuring Scheduled Jobs for MSSQL](#)

3.4.1 Scheduled Jobs for Lookup Field Synchronization for MSSQL

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

The following scheduled jobs are used for lookup field synchronization:

- DBUM MSSQL DB Names Lookup Reconciliation
- DBUM MSSQL Languages Lookup Reconciliation
- DBUM MSSQL Roles Lookup Reconciliation

You must specify values for the attributes of these scheduled jobs. [Table 3-17](#) describes the attributes of these scheduled jobs. The procedure to configure scheduled jobs is described later in the guide.

Table 3-17 Attributes of the Scheduled Jobs for Lookup Field Synchronization for MSSQL

Attribute	Description
Code Key Attribute	Enter the name of the connector or target system attribute that is used to populate the Code Key column of the lookup definition (specified as the value of the Lookup Name attribute). Sample value: <u> NAME </u> Note: Do not change the value of this attribute.
Decode Attribute	Enter the name of the connector or target system attribute that is used to populate the Decode column of the lookup definition (specified as the value of the Lookup Name attribute). Sample value: <u> NAME </u>
IT Resource Name	Enter the name of the IT resource for the target system installation from which you want to reconcile user records. Default value: MSSQL DB

Table 3-17 (Cont.) Attributes of the Scheduled Jobs for Lookup Field Synchronization for MSSQL

Attribute	Description
Lookup Name	<p>This attribute holds the name of the lookup definition that maps each lookup definition with the data source from which values must be fetched.</p> <p>Depending on the scheduled job you are using, the default values are as follows:</p> <ul style="list-style-type: none"> For DBUM MSSQL DB Names Lookup Reconciliation - <code>Lookup.DBUM.MSSQL.DBNames</code> For DBUM MSSQL Languages Lookup Reconciliation - <code>Lookup.DBUM.MSSQL.DefaultLang</code> For DBUM MSSQL Roles Lookup Reconciliation - <code>Lookup.DBUM.MSSQL.DBRoles</code>
Object Type	<p>Enter the type of object whose values must be synchronized.</p> <p>Depending on the scheduled job you are using, the default values are as follows:</p> <ul style="list-style-type: none"> For DBUM MSSQL DB Names Lookup Reconciliation - <code>__DBNAMES__</code> For DBUM MSSQL Languages Lookup Reconciliation - <code>__LANGUAGES__</code> For DBUM MSSQL Roles Lookup Reconciliation - <code>__ROLES__</code> <p>Note: Do not change the value of this attribute.</p>
Resource Object Name	<p>Enter the name of the resource object that is used for reconciliation.</p> <p>Default value: <code>MSSQL DB User</code></p>

3.4.2 Attributes of the Scheduled Jobs for Reconciliation of MSSQL User and Login Entities

The following scheduled jobs are used to reconcile user data in the target resource (account management) mode of the connector:

- For MSSQL user entity: DBUM MSSQL User Target Reconciliation
- For MSSQL login entity: DBUM MSSQL User Login Target Reconciliation

The following scheduled job is used to reconcile user data in the trusted source (identity management) mode of the connector:

DBUM MSSQL Trusted Reconciliation

[Table 3-18](#) describes the attributes of the scheduled jobs for user operations.

Table 3-18 Attributes of the Scheduled Jobs for Reconciliation for MSSQL User and Login Entities

Attribute	Description
Filter	<p>Expression for filtering records that must be reconciled by the scheduled job</p> <p>By default, the value of this attribute is empty.</p> <p>Sample value: <code>equalTo('__Name__', 'SEPT12USER1')</code></p> <p>See Performing Limited Reconciliation from MSSQL for the syntax of this expression.</p>

Table 3-18 (Cont.) Attributes of the Scheduled Jobs for Reconciliation for MSSQL User and Login Entities

Attribute	Description
Incremental Recon Attribute	Time stamp at which the last reconciliation run started Sample value: <code>lastModified</code> Note: Do not enter a value for this attribute. The reconciliation engine automatically enters a value for this attribute.
IT Resource Name	Name of the IT resource for the target system installation from which you want to reconcile user records Default value: <code>MSSQL DB</code>
Latest Token	This attribute is used for internal purposes. By default, this value is empty.
Object Type	Type of object you want to reconcile For trusted mode and login entity scheduled jobs: <code>USERLOGIN</code> Default value: <code>User</code>
Resource Object Name	Name of the resource object that is used for reconciliation For login entity scheduled jobs: <code>MSSQL DB User Login</code> For trusted mode scheduled jobs: <code>MSSQL UserLogin Trusted</code> For all other scheduled jobs: <code>MSSQL DB User</code>
Scheduled Task Name	Name of the scheduled job Note: For the scheduled job included with this connector, you must not change the value of this attribute. However, if you create a copy of the task, then you can enter the unique name for that scheduled job as the value of this attribute.

3.4.3 Attributes of the Scheduled Jobs for MSSQL Delete Operations

The following scheduled jobs are used to reconcile deleted user data in the target resource (account management) mode of the connector.

- For MSSQL user entity: `DBUM MSSQL Delete User Target Reconciliation`
- For MSSQL login entity: `DBUM MSSQL Delete User Login Target Reconciliation`

The following scheduled job is used to reconcile deleted user data in the trusted source (identity management) mode of the connector:

`DBUM MSSQL Delete Trusted Reconciliation`

[Table 3-19](#) describes the attributes of the scheduled jobs for delete operations.

Table 3-19 Attributes of the Scheduled Jobs for Delete Operations

Attribute	Description
IT Resource Name	Name of the IT resource for the target system installation from which you want to reconcile user records For DBUM MSSQL Delete Trusted Reconciliation, enter the name of the IT resource created for trusted source mode. For DBUM MSSQL Delete User Login Target Reconciliation: MSSQL DB For DBUM MSSQL Delete User Target Reconciliation: MSSQL DB
Object Type	Type of object you want to reconcile For DBUM MSSQL Delete Trusted Reconciliation: USERLOGIN For DBUM MSSQL Delete User Login Target Reconciliation: USERLOGIN For DBUM MSSQL Delete User Target Reconciliation: User
Resource Object Name	Name of the resource object that is used for reconciliation For DBUM MSSQL Delete Trusted Reconciliation: MSSQL UserLogin Trusted For DBUM MSSQL Delete User Login Target Reconciliation: MSSQL DB User Login For DBUM MSSQL Delete User Target Reconciliation: MSSQL DB User

3.4.4 Configuring Scheduled Jobs for MSSQL

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

See [Scheduled Jobs for Lookup Field Synchronization for MSSQL](#) and [Attributes of the Scheduled Jobs for MSSQL Delete Operations](#) for the scheduled jobs that are part of the connector and for information about their attributes.

To configure a scheduled job:

1. If you are using Oracle Identity Manager release 11.1.1.x:
 - a. Log in to the Administrative and User Console.
 - b. On the Welcome to Oracle Identity Manager Self Service page, click **Advanced** in the upper-right corner of the page.
 - c. On the Welcome to Oracle Identity Manager Advanced Administration page, in the System Management region, click **Search Scheduled Jobs**.
2. If you are using Oracle Identity Manager release 11.1.2.x or later:
 - a. Log in to Oracle Identity System Administration.
 - b. In the left pane, under System Management, click **Scheduler**.
3. Search for and open the scheduled job as follows:
 - a. On the left pane, in the Search field, enter the name of the scheduled job as the search criterion. Alternatively, you can click **Advanced Search** and specify the search criterion.
 - b. In the search results table on the left pane, click the scheduled job in the Job Name column.
4. On the Job Details tab, you can modify the following parameters:

Retries: Enter an integer value in this field. This number represents the number of times the scheduler tries to start the job before assigning the Stopped status to the job.

Schedule Type: Depending on the frequency at which you want the job to run, select the appropriate schedule type.

 **Note:**

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

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

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

 **Note:**

- Attribute values are predefined in the connector XML file that you import. Specify values only for those attributes that you want to change.
- Attributes of the scheduled job are discussed in [Attributes of the Scheduled Jobs for MSSQL Delete Operations](#).

6. After specifying the attributes, click **Apply** to save the changes.

3.5 Reconciliation from MSSQL

Reconciliation involves duplicating in Oracle Identity Manager the creation of and modifications to user accounts on the target system. This section discusses the following topics related to configuring reconciliation:

 **Note:**

Batched reconciliation is not supported in MSSQL target system.

- [Guidelines on Configuring Reconciliation from MSSQL](#)
- [About the Reconciliation Process for MSSQL](#)
- [About the Reconciliation Queries for MSSQL](#)
- [Target System Columns Used in Reconciliation from MSSQL](#)
- [Configuring the Target System As a Trusted Source](#)
- [Reconciliation Rules for MSSQL](#)
- [Viewing Reconciliation Rules for MSSQL](#)

- [Reconciliation Action Rules for MSSQL](#)
- [Viewing Reconciliation Action Rules for MSSQL](#)
- [Performing Full Reconciliation from MSSQL](#)
- [Performing Limited Reconciliation from MSSQL](#)
- [Performing Incremental Reconciliation from MSSQL](#)

3.5.1 Guidelines on Configuring Reconciliation from MSSQL

The following are guidelines that you must apply while configuring reconciliation:

- Before a target resource reconciliation run is performed, lookup definitions must be synchronized with the lookup fields of the target system. In other words, the scheduled job for lookup field synchronization must be run before user reconciliation runs.
- The scheduled job for user or login reconciliation must be run before the scheduled job for reconciliation of deleted user or login data.

3.5.2 About the Reconciliation Process for MSSQL

This connector can be configured to perform either trusted source reconciliation or target resource reconciliation.

When you configure the target system as a target resource, the connector enables you to create and manage database accounts for OIM Users through provisioning. In addition, data related to newly created and modified target system accounts can be reconciled and linked with existing OIM Users and provisioned resources.

When you configure the target system as a trusted source, the connector fetches into Oracle Identity Manager, data about newly created or modified target system accounts. This data is used to create or update OIM Users. See [Configuring the Target System As a Trusted Source](#) for more information.

See Also:

Reconciliation Based on the Object Being Reconciled in *Oracle Fusion Middleware Administering Oracle Identity Manager* for conceptual information about target resource reconciliation and trusted source reconciliation.

The following is an overview of the steps involved in reconciliation:

1. Depending on the target system that you are using, a SQL query or stored procedure is used to fetch target system records during reconciliation.
2. The scheduled job communicates to connector bundle and runs search operations over it, maps the task attributes to parameters of the reconciliation query or stored procedure, and then runs the query or stored procedure on the target system.
3. Target system records that meet the query or stored procedure criteria are fetched into Oracle Identity Manager.
4. If you have configured your target system as a trusted source, then each user record fetched from the target system is compared with existing OIM Users. The reconciliation rule is applied during the comparison process.

The next step of the process depends on the outcome of the matching operation:

- If a match is found between the target system record and the OIM User, then the OIM User attributes are updated with changes made to the target system record.
 - If no match is found, then the target system record is used to create an OIM User.
5. If you have configured your target system as a target resource, then each user record fetched from the target system is compared with existing target system resources assigned to OIM Users. The reconciliation rule is applied during the comparison process.

The next step of the process depends on the outcome of the matching operation:

- If a match is found between the target system record and a resource provisioned to an OIM User, then the database user resource is updated with changes made to the target system record.
- If no match is found, then the target system user record is compared with existing OIM Users. The next step depends on the outcome of the matching operation:

If a match is found, then the target system record is used to provision a resource for the OIM User.

If no match is found, then the status of the reconciliation event is set to No Match Found.



Note:

[Reconciliation Rules for MSSQL](#) for information about the reconciliation rule

3.5.3 About the Reconciliation Queries for MSSQL

As mentioned earlier in this chapter, a SQL query or a stored procedure is used to fetch target system records during reconciliation. All predefined SQL queries and stored procedures are stored in a JAR file in the bundle directory of the connector installation media.

For example, to locate the reconciliation query file, you can extract the `bundle/org.identityconnectors.dbum-1.0.1116.jar` file and open `scripts/mssql/Search.queries`.



Note:

Depending on your requirements, you can modify existing queries or add your own query in the query file. Alternatively, you can create and use your own query file. [Modifying Predefined Queries or Creating New Queries for MSSQL](#) provides more information.

Some of the predefined queries for MSSQL are used in conjunction with the Incremental Recon Attribute scheduled job attribute. This attribute stores the time stamp at which the last reconciliation run started. When the next reconciliation run begins, only target system records for which the lastModified column value is greater than the value of the Incremental Recon Attribute are fetched into Oracle Identity Manager. In other words, only records that were added or modified after the last reconciliation run started are considered for the current reconciliation run.

The following are the predefined queries for MSSQL:

- **SEARCH_USER**
This query is used to fetch user records and their lastModified time-stamps using the sp_helpuser() function.
- **SEARCH_USERLOGIN**
This query is used to fetch user records from the sys.syslogins table.
- **LOGIN_DATA_QUERY**
This stored procedure is used to fetch user data using the sp_helplogins() function.
- **LOGIN_STATUS_AUTH_QUERY**
This query is used to fetch user authentication data from the sys.server_principals table.
- **USER_DATA_QUERY**
This stored procedure is used to fetch user records using the sp_helpuser() function.

3.5.4 Target System Columns Used in Reconciliation from MSSQL

As mentioned earlier in this guide, this connector can be configured to perform either target resource reconciliation or trusted source reconciliation. This section discusses the following topics:

- **For Microsoft SQL Server login entity**
The Lookup.DBUM.MSSQL.Login.ReconAttrMap lookup definition holds attribute mappings for login data reconciliation. See [Lookup Definitions for Attribute Mappings for MSSQL Login Entity](#) for more information about this lookup definition.
- **For Microsoft SQL Server user entity**
The Lookup.DBUM.MSSQL.UM.ReconAttrMap and Lookup.DBUM.MSSQL.UM.ReconAttrMap.Trusted lookup definitions hold the attribute mappings for user data reconciliation. See [Lookup Definitions for Attribute Mappings for MSSQL User Entity](#) for more information about this lookup definition.

3.5.5 Configuring the Target System As a Trusted Source

**Note:**

Skip this section if you do not want to designate the target system as a trusted source for reconciliation.

To configure trusted source reconciliation:

1. If you are using Oracle Identity Manager release 11.1.1.x:
 - a. Log in to the Administrative and User Console.
 - b. On the Welcome page, click **Advanced** in the upper-right corner of the page.
 - c. On the Welcome to Oracle Identity Manager Advanced Administration page, in the Configuration region, click **Create IT Resource**.
 2. If you are using Oracle Identity Manager release 11.1.2.x or later:
 - a. Log in to Oracle Identity System Administration.
 - b. In the left pane, under Configuration, click **IT Resource**.
 - c. In the Manage IT Resource page, click **Create IT Resource**.
 3. On the Step 1: Provide IT Resource Information page, enter the following information:
 - **IT Resource Name:** Enter a name for the IT resource. For example, `MSSQL_DB Trusted`.
 - **IT Resource Type:** Select the **MSSQL DB** IT resource type for the IT resource.
 4. Click **Continue**.
 5. On the Step 2: Specify IT Resource Parameter Values page, specify values for the parameters of the IT resource.

Configuration Lookup: Name of the lookup definition in which you store the connector configuration information for the target system.

Sample Value: `Lookup.DBUM.MSSQL.Configuration.Trusted`

Provide values for the other IT resource parameters.
 6. Click **Continue**.
- In the following steps, provide permissions on the IT resource that you are creating as per your requirements.

You can use this IT resource for trusted source reconciliation operations.

3.5.6 Reconciliation Rules for MSSQL



See Also:

Reconciliation Metadata in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for generic information about reconciliation rules and reconciliation action rules

The following reconciliation rules are used by the reconciliation engine for this connector:

- **Rule name for user entity in target resource mode:** DBUM MSSQL User Target Recon
- Rule element:** User Login Equals User Name
- In this rule:

- User Login is the field on the OIM User form.
- User Name is the target system field.
- **Rule name for login entity in target resource mode:** DBUM MSSQL Login Target Recon

Rule element: User Login Equals Login Name

In this rule element:

 - User Login is the field on the OIM User form.
 - Login Name is the target system field.
- **Rule name for trusted source mode:** MSSQL UserLogin Trusted

Rule element: User Login Equals User ID

In this rule element:

 - User Login is the field on the OIM User form.
 - User ID is the target system field.

3.5.7 Viewing Reconciliation Rules for MSSQL

After you deploy the connector, you can view the reconciliation rule for reconciliation by performing the following steps:



Note:

Perform the following procedure only after the connector is deployed.

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Development Tools**.
3. Double-click **Reconciliation Rules**.
4. Search for the rule name.

3.5.8 Reconciliation Action Rules for MSSQL

Reconciliation action rules define that actions the connector must perform based on the reconciliation rules defined for Users.

[Table 3-20](#) lists the action rules for target resource reconciliation.

Table 3-20 Action Rules for Target Resource Reconciliation for MSSQL

Rule Condition	Action
No Matches Found	Assign to Administrator With Least Load
One Entity Match Found	Establish Link
One Process Match Found	Establish Link

[Table 3-21](#) lists the action rules for trusted source reconciliation.

Table 3-21 Action Rules for Trusted Source Reconciliation for MSSQL

Rule Condition	Action
No Matches Found	Create User
One Entity Match Found	Establish Link

3.5.9 Viewing Reconciliation Action Rules for MSSQL

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

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Resource Management**.
3. Double-click **Resource Objects**.
4. Search for and open the resource object. The following are the names of the resource objects for each target system database:
 - Resource object for MSSQL user entity:
MSSQL DB User
 - Resource object for MSSQL login entity:
MSSQL DB User Login
 - Resource object for MSSQL as trusted source:
MSSQL UserLogin Trusted
5. Click the **Object Reconciliation** tab, and then click the **Reconciliation Action Rules** tab. The Reconciliation Action Rules tab displays the action rules defined for this connector.

3.5.10 Performing Full Reconciliation from MSSQL

Full reconciliation involves reconciling all existing user records from the target system into Oracle Identity Manager. After you deploy the connector, you must first perform full reconciliation.

To perform a full reconciliation run, remove (delete) any value currently assigned to the Filter attribute and run one of the following scheduled jobs:

- For MSSQL user entity in target resource mode: DBUM MSSQL User Target Reconciliation
- For MSSQL login entity in target resource mode: DBUM MSSQL User Login Target Reconciliation
- For MSSQL as a trusted source: DBUM MSSQL Trusted Reconciliation

See [Attributes of the Scheduled Jobs for MSSQL Delete Operations](#) for information about this scheduled job.

3.5.11 Performing Limited Reconciliation from MSSQL

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

You can perform limited reconciliation by creating filters for the reconciliation module. This connector provides a Filter attribute (a scheduled task attribute) that allows you to use any of the DBUM resource attributes to filter the target system records. You can apply filters to the parent parameters in the reconciliation query file stored in a JAR file in the bundle directory of the connector installation media. For example, to locate the reconciliation query file, you can extract the `bundle/org.identityconnectors.dbum-1.0.1116.jar` file and open `scripts/mssql/Search.queries`.

The parent parameters that can be used with the Filter attribute of the scheduled jobs are `__UID__` and `__NAME__`.

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

While deploying the connector, follow the instructions in [Configuring Scheduled Jobs for MSSQL](#) to specify attribute values.

3.5.12 Performing Incremental Reconciliation from MSSQL

During an incremental reconciliation run, the scheduled job fetches only target system records that are added or modified after the time-stamp stored in the Latest Token attribute of the scheduled job. The connector requires a query to calculate the time-stamp value. This time-stamp value is used by the query that is used to perform reconciliation.

There is no configuration required for the connector to perform incremental reconciliation. After you run full reconciliation, the latest time-stamp of all the retrieved records is updated in the scheduled job and is used for next reconciliation runs.

3.6 Provisioning for MSSQL

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

This section contains the following topics about provisioning:

- [Guidelines on Performing Provisioning Operations for MSSQL](#)
- [Understanding the Provisioning Process for MSSQL](#)
- [Configuring Direct Provisioning for MSSQL](#)
- [Configuring Request-Based Provisioning for MSSQL](#)
- [Switching Between Request-Based Provisioning and Direct Provisioning for MSSQL](#)
- [Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x](#)

3.6.1 Guidelines on Performing Provisioning Operations for MSSQL

The following are guidelines that you must apply while performing provisioning operations:

- Before you perform provisioning operations, lookup definitions must be synchronized with the lookup fields of the target system. In other words, run the scheduled jobs for lookup field synchronization before provisioning operations.
- The character length of target system fields must be taken into account when specifying values for the corresponding Oracle Identity Manager fields.
- During an update password provisioning operation, ensure that you clear the existing text in the Password field, and then enter the new password.
- Before you provision a Microsoft SQL Server account that uses Windows Authentication, you must ensure that the account you want to provision exists in the account database of the operation system.
- If you are creating users accounts, then you must specify a value for the Database Name parameter of the IT resource.
- If you are provisioning a Microsoft SQL Server login account that uses Windows Authentication, then you must specify values for the following fields:
 - **Default Database:** Select the name of the default database that the user must connect to.
 - **Default Language:** Select the default language for the login.
 - **Login Name:** Enter the login name in the following format:
`[DOMAIN_NAME\LOGIN_NAME]`
In this format:
 - * `DOMAIN_NAME` is the name of the domain to which the login account must belong.
 - * `LOGIN_NAME` is the name of the login that you are creating in the target system.The following is a sample value that you can enter in the Login Name field:
`[MyDomain\jdoe]`
- If you are provisioning a Microsoft SQL Server login account that uses SQL Server Authentication, then you must specify values for the following mandatory fields:
 - **Login Name:** Enter the name of the login account.
 - **Password:** Enter the password for the login account.

 **Note:**

Microsoft SQL Server has a strict password policy. However, the connector does not validate this password which may result in some failed operations.

You can add custom validation to ensure the default password policy of the target system is followed. If you do so, you must configure the validation for the password field in the Lookup.DBUM.MSSQL.Login.ProvValidations lookup definition.

3.6.2 Understanding the Provisioning Process for MSSQL

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

When you install the connector on Oracle Identity Manager, the direct provisioning feature is automatically enabled. This means that the process form is enabled when you install the connector.

The following are types of provisioning operations:

- Direct provisioning
- Request-based provisioning
- Provisioning triggered by policy changes

If you configure the connector for request-based provisioning, then the process form is suppressed and the object form is displayed. In other words, direct provisioning is disabled when you configure the connector for request-based provisioning. If you want to revert to direct provisioning, then see [Switching Between Request-Based Provisioning and Direct Provisioning for MSSQL](#).

The following is an overview of the Create User provisioning process in MSSQL that is started through direct provisioning:

1. On the Create User page of the Administrative and User Console, the administrator enters the data required for an OIM User account creation.

Suppose the administrator enters the following values for the fields on the Create User page:

- First Name: John
- Last Name: Doe
- User ID: jdoe

An OIM User account is created for John Doe.

2. The administrator selects the resource to be provisioned to the OIM User account that has been created.

Before provisioning a MSSQL User resource, you must first provision a MSSQL Userlogin resource on the same target system. During this provisioning, you have to provide login ID for the user.

In this example, the administrator selects the MSSQL DB User Login resource first and then selects the MSSQL DB User resource.

3. The administrator enters the data required for provisioning the MSSQL DB User resource. Suppose the administrator wants to create a local user that requires a password to log in to the database. Therefore, the administrator enters the following values on the resource provisioning process form:

- IT Resource: MSSQL DB User
- Login Name: JDoe
- Authentication Type: SQL_SERVER_AUTHENTICATION

- Password: my_pa55word
- Default DataBase: sqlldb1
- Default Language: example

In addition, the administrator also enters the following value on the process form for granting roles:

- Role: 3~db_datareader

You can grant roles for only a user entity and not for a login entity.

4. From the information available in the IT resource for the target system, the configuration (Lookup.DBUM.MSSQL.Configuration) lookup definition is identified. This lookup definition stores configuration information that is used during connector operations.
5. The connector bundle contains the script (Provisioning.queries) required for provisioning operations.
6. The identifiers in the SQL statement are replaced with the input parameters fetched from the query. Then, the SQL statement with actual values is formed.
7. The connector runs the SQL statement on MSSQL and creates the jdoe account on the target system.

If the administrator did not enter any values for granting roles, then the provisioning process ends here. Otherwise, the process continues to the next step.

8. While performing Step 3, the administrator had entered the required data for granting roles to the jdoe account. Therefore, the corresponding query as mentioned in Step 6 is fetched.
9. The complete SQL statement that must be run to perform the Add role provisioning operation is formed.
10. The input parameters required to run the SQL statement are fetched from the query file.
11. The identifiers in the SQL statement (formed in Step 9) are replaced with the input parameters fetched from the query. Then, the SQL statement with actual values is formed.
12. The query runs the SQL statement on the target system (MSSQL) and grants the role to the jdoe target system account.

3.6.3 Configuring Direct Provisioning for MSSQL

In direct provisioning, the Oracle Identity Manager administrator uses the Administrative and User Console to create a target system account for a user.

To provision a resource by using the direct provisioning approach:

1. Log in to the Administrative and User Console.
2. If you want to first create an OIM User and then provision a database account to the user, then:
 - a. On the Welcome to Identity Administration page, in the Users region, click **Create User**.
 - b. On the Create User page, enter values for the OIM User fields, and then click the save icon.

3. If you want to provision a database account to an existing OIM User, then:
 - a. On the Welcome to Identity Administration page, search for the user by selecting **Users** from the Search list on the left pane.

Alternatively, in the Users region, click **Advanced Search - User**, provide a search criterion, and then click **Search**.
 - b. From the list of users displayed in the search results, select the OIM User.

The user details page is displayed.
4. From the Action menu, select **Add Resource**. Alternatively, you can click the add resource icon with the plus (+) sign. The Provision Resource to User page is displayed in a new window.
5. On the Step 1: Select a Resource page, select MSSQL DB User Login or MSSQL DB User as the resource from the list, and then click **Continue**.

You must first provision the MSSQL DB User Login resource to a user before provisioning the MSSQL DB User resource.
6. On the Step 2: Verify Resource Selection page, click **Continue**.
7. On the Step 5: Provide Process Data page, enter the details of the account that you want to create on the target system and then click **Continue**.
8. If you want to provide child data, then on the Step 5: Provide Process Data page for child data, search for and select the child data for the user on the target system and then click **Continue**. Repeat the same step if you have more than one child data and you want to provision them.
9. On the Step 6: Verify Process Data page, verify the data that you have provided and then click **Continue**.
10. The "Provisioning has been initiated" message is displayed. Perform the following step:
 - a. Close the window displaying the "Provisioning has been initiated" message.
 - b. On the Resources tab, click **Refresh** to view the newly provisioned resource.
If the resource status is Provisioned, then provisioning was successful. If the status is Provisioning, then there may be an error. To verify if there was an error, you can check the resource history.

3.6.4 Configuring Request-Based Provisioning for MSSQL

The following sections discuss the steps to be performed to enable request-based provisioning:

- [About Request Based Provisioning for MSSQL](#)
- [Enabling Request-Based Provisioning](#)

3.6.4.1 About Request Based Provisioning for MSSQL

In request-based provisioning, an end user creates a request for a resource by using the Administrative and User Console. Administrators or other users can also create requests for a particular user. Requests for a particular resource on the resource can be viewed and approved by approvers designated in Oracle Identity Manager.

The following are features of request-based provisioning:

- A user can be provisioned only one resource (account) on the target system.

 **Note:**

Direct provisioning allows the provisioning of multiple database accounts on the target system.

- Direct provisioning cannot be used if you enable request-based provisioning.

3.6.4.2 Enabling Request-Based Provisioning

The following sections provide information about the procedures you must perform to enable request-based provisioning:

 **Note:**

The procedure described in this section is applicable only if you are using Oracle Identity Manager release 11.1.1.x.

- [Approver's Role in Request-Based Provisioning for MSSQL](#)
- [Importing MSSQL Request Datasets Using Deployment Manager](#)
- [End User's Role in Request-Based Provisioning for MSSQL](#)
- [Enabling the Auto Save Form Feature for MSSQL](#)
- [Running the PurgeCache Utility for MSSQL](#)

3.6.4.2.1 Approver's Role in Request-Based Provisioning for MSSQL

The following are steps performed by the approver in a request-based provisioning operation:

1. Log in to the Administrative and User Console.
2. On the Welcome page, click **Self-Service** in the upper-right corner of the page.
3. On the Welcome to Identity Manager Self Service page, click the **Tasks** tab.
4. On the **Approvals** tab, in the first section, you can specify a search criterion for request task that is assigned to you.
5. From the search results table, select the row containing the request you want to approve, and then click **Approve Task**.

A message confirming that the task was approved is displayed.

3.6.4.2.2 Importing MSSQL Request Datasets Using Deployment Manager

A request dataset is an XML file that specifies the information to be submitted by the requester during a provisioning operation. These request datasets specify information about the default set of attributes for which the requester must submit information during a request-based provisioning operation.

To import a request dataset XML file by using the Deployment Manager:

1. Log in to the Oracle Identity Manager Administrative and User Console.
2. Click the **Deployment Management** link on the left navigation bar.
3. Click the **Import** link under Deployment Management.
A dialog box for opening files is displayed.
4. Locate and open the request dataset XML file, DBUserManagement-MSSQL-Datasets.xml, which is in the xml directory of the installation media.
Details of this XML file are shown on the **File Preview** page.
5. Click **Add File**.
The Substitutions page is displayed.
6. Click **Next**.
The Confirmation page is displayed.
7. Click **Import**.
8. Close the Deployment Manager dialog box.
The request dataset is imported into Oracle Identity Manager.

3.6.4.2.3 End User's Role in Request-Based Provisioning for MSSQL

The following steps are performed by the end user in a request-based provisioning operation:

1. Log in to the Administrative and User Console.
2. On the Welcome page, click **Advanced** in the upper-right corner of the page.
3. On the Welcome to Identity Administration page, click the **Administration** tab, and then click the **Requests** tab.
4. From the Actions menu on the left pane, select **Create Request**.
The Select Request Template page is displayed.
5. From the Request Template list, select **Provision Resource** and click **Next**.
6. On the Select Users page, specify a search criterion in the fields to search for the user that you want to provision the resource, and then click **Search**. A list of users that match the search criterion you specify is displayed in the Available Users list.
7. From the **Available Users** list, select the user to whom you want to provision the account.
If you want to create a provisioning request for more than one user, then from the **Available Users** list, select users to whom you want to provision the account.
8. Click **Move** or **Move All** to include your selection in the Selected Users list, and then click **Next**.
9. On the Select Resources page, click the arrow button next to the Resource Name field to display the list of all available resources.
10. From the Available Resources list, select **MSSQL DB User Login** and **MSSQL DB User**, move them to the Selected Resources list, and then click **Next**.
11. On the Resource Details page, enter details of the account that must be created on the target system, and then click **Next**.

12. On the Justification page, you can specify values for the following fields, and then click **Finish**.
 - Effective Date
 - JustificationA message confirming that your request has been sent successfully is displayed along with the Request ID.
13. If you click the request ID, then the Request Details page is displayed.
14. To view details of the approval, on the Request Details page, click the **Request History** tab.

3.6.4.2.4 Enabling the Auto Save Form Feature for MSSQL

To enable the Auto Save Form feature:

1. Log in to the Design Console.
2. Expand **Process Management**, and then double-click **Process Definition**.
3. Search for and open the **MSSQL DB User** process definition.
4. Select the **Auto Save Form** check box.
5. Click the save icon.

3.6.4.2.5 Running the PurgeCache Utility for MSSQL

Run the PurgeCache utility to clear content belonging to the Metadata category from the server cache.

See [Clearing Content Related to the Connector Resource Bundles from the Server Cache](#) for instructions.

The procedure to enable enabling request-based provisioning ends with this step.

3.6.5 Switching Between Request-Based Provisioning and Direct Provisioning for MSSQL

If you have configured the connector for request-based provisioning, you can always switch to direct provisioning. Similarly, you can always switch back to request-based provisioning any time. This section discusses the following topics:

- [Switching From Request-Based Provisioning to Direct Provisioning](#)
- [Switching From Direct Provisioning to Request-Based Provisioning](#)

3.6.5.1 Switching From Request-Based Provisioning to Direct Provisioning



Note:

It is assumed that you have performed the procedure described in [Configuring Request-Based Provisioning for MSSQL](#).

If you want to switch from request-based provisioning to direct provisioning, then:

1. Log in to the Design Console.
2. Disable the Auto Save Form feature as follows:
 - a. Expand **Process Management**, and then double-click **Process Definition**.
 - b. Search for and open the **MSSQL DB** process definition.
 - c. Deselect the Auto Save Form check box.
 - d. Click the save icon.
3. If the Self Request Allowed feature is enabled, then:
 - a. Expand **Resource Management**, and then double-click **Resource Objects**.
 - b. If you have configured the target system as a trusted source, then search for and open the **MSSQL UserLogin Trusted** resource object.
 - c. If you have configured the target system as a target resource, then search for and open the **MSSQL DB User** (user entity) or **MSSQL DB User Login** (login entity) resource object.
 - d. Deselect the Self Request Allowed check box.
 - e. Click the save icon.

3.6.5.2 Switching From Direct Provisioning to Request-Based Provisioning

If you want to switch from direct provisioning back to request-based provisioning, then:

1. Log in to the Design Console.
2. Enable the Auto Save Form feature as follows:
 - a. Expand **Process Management**, and then double-click **Process Definition**.
 - b. Search for and open the **MSSQL DB** process definition.
 - c. Select the **Auto Save Form** check box.
 - d. Click the save icon.
3. If you want to enable end users to raise requests for themselves, then:
 - a. Expand **Resource Management**, and then double-click **Resource Objects**.
 - b. If you have configured the target system as a trusted source, then search for and open the **MSSQL UserLogin Trusted** resource object.
 - c. If you have configured the target system as a target resource, then search for and open the **MSSQL DB User** (user entity) or **MSSQL DB User Login** (login entity) resource object.
 - d. Select the Self Request Allowed check box.
 - e. Click the save icon.

3.6.6 Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x

To perform provisioning operations in Oracle Identity Manager release 11.1.2.x:

1. Log in to Identity Self Service.

2. If you want to first create an OIM User and then provision a target system account, then:

 **Note:**

See *Creating Users in Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Manager* for more information about creating a user.

- a. In the left pane, under Administration, click **Users**.
The Search Users page is displayed.
 - b. From the Actions menu, select **Create**. Alternatively, you can click **Create** on the toolbar.
 - c. On the Create User page, enter values for the OIM User fields, and then click **Submit**. A message is displayed stating that the user is created successfully.
3. If you want to provision a target system account to an existing OIM User, then:

 **Note:**

See *Searching Users in Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Manager* for more information about searching a user.

- a. In the left pane, under Administration, click **Users**.
The Search Users page is displayed.
 - b. Specify a search criteria to search for the OIM User, and then click **Search**.
 - c. From the list of users displayed in the search results, select the OIM User. The user details page is displayed on the right pane.
4. On the Account tab, click **Request Accounts**.
 5. In the Catalog page, search for and add to cart the application instance (in other words, the account to be provisioned), and then click **Checkout**.
 6. Specify value for fields in the application form and then click **Ready to Submit**.
 7. Click **Submit**.
 8. If you want to provision entitlements, then:
 - a. On the Entitlements tab, click **Request Entitlements**.
 - b. In the Catalog page, search for and add to cart the entitlement, and then click **Checkout**.
 - c. Click **Submit**.

3.7 Extending the Connector for MSSQL

The following sections describe procedures that you can perform to extend the functionality of the connector for addressing your specific business requirements:



Note:

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

- [Modifying Predefined Queries or Creating New Queries for MSSQL](#)
- [Syntax of Provisioning Queries for MSSQL](#)
- [Configuring Queries to Add Support for Custom Parameters and Lookup Fields for MSSQL](#)
- [About Configuring the Connector for Multiple Installations of MSSQL](#)
- [About Configuring the Connector for Multiple Trusted Source Reconciliation from MSSQL](#)
- [Configuring Validation of Data During Reconciliation and Provisioning for MSSQL](#)
- [Configuring Transformation of Data During User Reconciliation for MSSQL](#)
- [Configuring Resource Exclusion Lists for MSSQL](#)
- [Setting Up Action Scripts for MSSQL](#)

3.7.1 Modifying Predefined Queries or Creating New Queries for MSSQL

The following sections discuss syntax and guidelines that you must apply while modifying the predefined queries or creating new queries:

- [About the Queries for MSSQL](#)
- [Syntax of Provisioning Queries for MSSQL](#)
- [Syntax of Reconciliation Queries for MSSQL](#)
- [Syntax of List of Values Queries for MSSQL](#)
- [Guidelines for Configuring Search Queries Used in Reconciliation from MSSQL](#)

3.7.1.1 About the Queries for MSSQL

Predefined queries are provided to reconcile target system user records, synchronize lookup field values with Oracle Identity Manager, and for provisioning operations. You can modify the predefined queries or add your own queries.

The query files are included in a JAR file in the bundle directory of the connector installation media. For example, `bundle/org.identityconnectors.dbum-1.0.1116.jar`.

The connector includes the following types of queries:

- **Provisioning Queries**
They are used for create, update, and delete operations. The query file is `scripts/mssql/Provisioning.queries`.
- **List of Values Search Queries**
They are used for reconciliation of lookup definitions. A list of value query operates on a set of values for fields such as database names, roles, and languages. The query file is `scripts/mssql/LoVSearch.queries`.
- **Account Search Queries**
They are used for full, incremental, and delete reconciliation operations. An account search query operates on account and group searches with various conditions. The query file is `scripts/mssql/Search.queries`.

**Note:**

The stored procedure OUT parameters cannot be configured for write-back on the process form. The returned values cannot be used for any connector operations.

3.7.1.2 Syntax of Provisioning Queries for MSSQL

The following is the syntax of the queries used for provisioning operations:

```

QUERYID {
  Query="QUERY"
  QueryType="QUERYTYPE"
  Parameters=["PARAM1":"PARAMDEFN1", "PARAM2":"PARAMDEFN2"...]
  ExtensionJoin="EXTENSIONJOIN"
  ExtensionSeparator="EXTENSIONSEPARATOR"
  QueryExtensions=["EXTENSION1", "EXTENSION2"...]
}

```

For example:

```

CREATE_SQL_SERVER_AUTHENTICATION_USERLOGIN {
  Query="CREATE LOGIN {__NAME__} WITH PASSWORD={__PASSWORD__}"
  QueryType="SQL"
  Parameters=["__NAME__": "Type:String", "__PASSWORD__": "Type:GuardedString",
  Tags:QUOTES"]
  ExtensionJoin=", "
  ExtensionSeparator=", "
  QueryExtensions=["DEFAULT_DATABASE", "DEFAULT_LANGUAGE"]
}

```

In this syntax:

- **QUERYID** refers to the unique name of the query.
For example: `CREATE_SQL_SERVER_AUTHENTICATION_USERLOGIN`

For CREATE provisioning queries, the format of *QUERYID* is *CREATE_AUTHENTICATIONTYPE_ENTITY*. The format for other provisioning queries can be *OPERATIONTYPE_ENTITY* (such as *DELETE_USERLOGIN*), *ENTITY_OPERATIONTYPE_ATTRIBUTE* (such as *USERLOGIN_UPDATE_DEFAULTDATABASE*) or *OPERATIONTYPE_OPERATION* (such as *UPDATE_ADD_ROLES*).

- *QUERY* refers to the main query.

For example: `Query="CREATE LOGIN {__NAME__} WITH PASSWORD={__PASSWORD__}"`

- *QueryType* refers to the type of the main query, either an SQL query, a stored procedure, or a query extension. The value of *QUERYTYPE* can be *SQL*, *StoredProc*, or *QUERYEXTENSION*.

For example: `QueryType="SQL"`

- *Parameters* refers to the list of comma separated parameters and parameter definitions used with the main query, represented by "*PARAM1*":"*PARAMDEFN1*", "*PARAM2*":"*PARAMDEFN2*", and so on.

For example:

```
Parameters=["__NAME__":"Type:String", "__PASSWORD__":"Type:GuardedString",
Tags:QUOTES"]
```

A parameter can have the following attributes:

- *Type* is the type of the parameter.
- *Direction* is the flow of data from the query to or from the parameter. It can have a value of *IN*, *OUT*, or *INOUT*.
- *TAGS* is the enclosure characters that are applied to the parameter before the query is processed. It can have a value of *DOUBLEQUOTES*, *QUOTES*, *UPPERCASE*, or *LOWERCASE*. You can use multiple tags in double quotes and separated by commas. However, you must not use *DOUBLEQUOTES* with *QUOTES* or *UPPERCASE* with *LOWERCASE* in the same query.
- *ExtensionJoin* (optional) refers to the operator, represented by *EXTENSIONJOIN*, used to join the main query with query extensions.

For example: `ExtensionJoin=", "`

- *ExtensionSeparator* (optional) refers to the delimiter between query extensions, represented by *EXTENSIONSEPARATOR*.

For example: `ExtensionSeparator=", "`

- *QueryExtensions* (optional) refers to the extensions that must be appended to the main query, represented by *EXTENSION1*, *EXTENSION2*, and so on.

For example:

```
QueryExtensions=["DEFAULT_DATABASE", "DEFAULT_LANGUAGE"]
```

During an operation, the connector combines all these components to the following query:

```
QUERY PARAM1, PARAM2... [EXTENSIONJOIN [EXTENSION1 EXTENSIONSEPARATOR
EXTENSION2 EXTENSIONSEPARATOR...]]
```

For example:

```
CREATE LOGIN {__NAME__} WITH PASSWORD={__PASSWORD__},
DEFAULT_DATABASE={defaultDatabase}, DEFAULT_LANGUAGE={defaultLanguage}
```

Table 3-22 lists the script selection logic of the provisioning queries:

Table 3-22 Script Section Logic for MSSQL Provisioning Queries

Operation	Selection Logic	Query IDs
CREATE	CREATE_AUTHTYPE_OBJECTTYPE CREATE_OBJECTTYPE	CREATE_SQL_SERVER_AUTHENTICATION_USERLOGIN CREATE_WINDOWS_AUTHENTICATION_USERLOGIN CREATE_USER
DELETE	DELETE_OBJECTTYPE	DELETE_USERLOGIN DELETE_USER
ENABLE	ENABLE_OBJECTTYPE	ENABLE_USERLOGIN
DISABLE	DISABLE_OBJECTTYPE	DISABLE_USERLOGIN
RESET PASSWORD	OBJECTTYPE_SET_PASSWORD	USERLOGIN_SET_PASSWORD
UPDATE	OBJECTTYPE_UPDATE_ATTRIBUTE	USERLOGIN_UPDATE_DEFAULTDATABASE USERLOGIN_UPDATE_DEFAULTLANGUAGE USERLOGIN_UPDATE_LOGINNAME
ADD CHILD VALUES	UPDATE_ADD_ATTRIBUTE	UPDATE_ADD_ROLES
REMOVE CHILD VALUES	UPDATE_REVOKE_ATTRIBUTE	UPDATE_REVOKE_ROLES

3.7.1.3 Syntax of Reconciliation Queries for MSSQL

The following is the syntax of the search queries used during reconciliation operations:

```

QUERYID {
  Query="QUERY"
  QueryType="QUERYTYPE"
  Parameters=["PARAM1":"PARAMDEFN1", "PARAM2":"PARAMDEFN2"...]
  ExtensionJoin="EXTENSIONJOIN"
  ExtensionSeparator="EXTENSIONSEPARATOR"
  QueryExtensions=["EXTENSION1", "EXTENSION2"...]
}

```

For example:

```

SEARCH_USER {
  Query="select {__UID__}, {lastModified} from (select * from sys.sysusers
where issqlrole=0) OUTERQUERY {filter}"
  QueryType="SQL"
  Parameters=["__UID__": "Type:String, Direction:OUT, ColName:name",
"lastModified": "Type:long, Direction:OUT, ColName:TS,

```

```
ColQuery:\\"cast((cast(updatedate as float)*100000) as bigint)\\""]
  QueryExtensions=["USER_DATA_QUERY"]
}
```

In this syntax:

- *QUERYID* refers to the unique name of the query.

For example: SEARCH_USER

QUERYID can be one of the following values:

- SEARCH_USER
- SEARCH_USERLOGIN

- Query refers to the main query, represented by *QUERY*.

For example: Query="select {__UID__}, {lastModified} from (select * from sys.sysusers where issqlrole=0) OUTERQUERY {filter}"

- QueryType refers to the type of the main query, either an SQL query, a stored procedure, or a query extension. The value of *QUERYTYPE* can be SQL, StoredProc, or QUERYEXTENSION.

For example: QueryType="SQL"

- Parameters refers to the list of comma separated parameters and parameter definitions used with the main query, represented by "*PARAM1*":"*PARAMDEFN1*", "*PARAM2*":"*PARAMDEFN2*", and so on.

For example:

```
Parameters=["__UID__": "Type:String,Direction:OUT,ColName:name",
"lastModified": "Type:long,Direction:OUT,ColName:TS",
ColQuery:\\"cast((cast(updatedate as float)*100000) as bigint)\\""]
```

A parameter can have the following attributes:

- Type is the type of the parameter.
- Direction is the flow of data from the query to or from the parameter. It can have a value of IN, OUT, or INOUT.
- ColName is the column name in the target system corresponding to the parameter in the query.
- ColQuery is the query used to fetch values for the corresponding query parameter.
- ExtensionJoin (optional) refers to the operator, represented by *EXTENSIONJOIN*, used to join the main query with query extensions.

For example: ExtensionJoin=","

- ExtensionSeparator (optional) refers to the delimiter between query extensions, represented by *EXTENSIONSEPARATOR*.

For example: ExtensionSeparator=","

- QueryExtensions (optional) refers to the extensions that must be appended to the main query, represented by *EXTENSION1*, *EXTENSION2*, and so on.

For example: QueryExtensions=["USER_DATA_QUERY"]

QueryExtensions can be one of the following:

- LOGIN_DATA_QUERY

- LOGIN_STATUS_AUTH_QUERY
- USER_DATA_QUERY

During a reconciliation operation, the connector combines all these components to the following query:

```
QUERY PARAM1, PARAM2... [EXTENSIONJOIN [EXTENSION1
EXTENSIONSEPARATOR EXTENSION2 EXTENSIONSEPARATOR...]]
```

For example:

```
SELECT {__UID__}, {lastModified} FROM sys.sysusers, USER_DATA_QUERY
```

3.7.1.4 Syntax of List of Values Queries for MSSQL

If a search query is performed on entities, such as user entity or login entity, then the query is considered as a reconciliation query. If a search query is performed on any other object, then the query is considered as a list of values query.

The following is the syntax of the list of values queries used for lookup field synchronization:

```
OBJECTTYPE = "QUERY"
```

For example:

```
__DBNAMES__="SELECT name FROM sys.sysdatabases"
```

In this syntax:

- *OBJECTTYPE* refers to the attribute provided by objectType scheduled job parameter.

For example: `__DBNAMES__`

- *QUERY* refers to the query used for fetching a lookup field attribute.

For example: `SELECT name FROM sys.sysdatabases`

The list of values queries return only one value, used as a lookup field value. By default, the connector includes a dedicated scheduled job for each lookup definition. To use a custom lookup definition, you must add custom fields in the query file.

3.7.1.5 Guidelines for Configuring Search Queries Used in Reconciliation from MSSQL

The following are guidelines that you must apply while modifying or creating queries for reconciliation:

- By adding or removing a column from the SELECT clause of a reconciliation query, you add or remove an attribute from the list of target system attributes for reconciliation. To enable the connector to process a change (addition or removal) in the list of reconciled attributes, you must make corresponding changes in the provisioning part of the connector.
- In the query properties file, you must not change the names of the predefined queries.
- Some of the predefined queries use inner queries. If you add or remove a column from the outer query, you must make corresponding changes in the inner queries.

- You must not remove columns corresponding to the Login Name and User Name resource object attributes.
- You must ensure that the following condition included in the Parameters list is not removed:

```
"lastModified":{"Type:long,Direction:IN,ColQuery:\"cast((cast(updatedate as float)*100000) as bigint)\"}]
```

This condition is used to determine if a target system record was added or updated after the time-stamp stored in the Incremental Recon Attribute scheduled job attribute.

- When you add or remove columns from the SELECT clause of the queries in the properties file, then you must update the attribute mapping lookup definition that holds mappings between child attributes and the target system column names. In addition, you must update other OIM objects.
- Before you modify or add a query in the Search.queries file, you must run the query by using any standard database client to ensure that the query produces the required results when it is run against the target system database.

3.7.2 Configuring Queries to Add Support for Custom Parameters and Lookup Fields for MSSQL

The connector uses preconfigured queries for connector operations such as create, delete, and search. You can add custom parameters and lookup definition fields as per your requirements.

The procedure to add a parameter or a lookup definition field to a query file is discussed in the following sections:

- [Updating a Query File for MSSQL](#)
- [Configuring Oracle Identity Manager](#)

3.7.2.1 Updating a Query File for MSSQL

To update a query file:

1. If the connector is already installed, run the Oracle Identity Manager Download JARs utility to download the connector bundle JAR file from the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

```
OIM_HOME/server/bin/DownloadJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/DownloadJars.sh
```

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

2. Copy the bundle JAR file in a temporary directory.

Sample JAR file: `bundle/org.identityconnectors.dbum-1.0.1116.jar`

Sample temporary directory: `c:\temp`

3. Run the following command to extract the manifest file, META-INF/MANIFEST.MF, from the JAR file:

```
jar -xvf org.identityconnectors.dbum-1.0.1116.jar
```

 **Note:**

You can also run the WinZip or WinRAR utility to extract the contents from the JAR file.

4. Delete the bundle JAR file in the temporary directory.
5. Update the value of **ConnectorBundle-Version** in the manifest file to a new value.

For example:

```
ConnectorBundle-Version: 1.0.1117
```

6. Depending on your requirement, update the query file with new parameters as per the query syntax described in [Modifying Predefined Queries or Creating New Queries for MSSQL](#).

For example, if you want to add a new parameter, `defaultLanguage`, to the `CREATE_SQL_SERVER_AUTHENTICATION_USERLOGIN` provisioning query:

- a. Open the query file in a text editor.

Sample query file:

```
c:\temp\bundle\org.identityconnectors.dbum-1.0.1116\scripts\mssql\Provisioning.queries
```

- b. Add the parameter, `defaultLanguage`, to the `CREATE_SQL_SERVER_AUTHENTICATION_USERLOGIN` query.

Sample updated query:

```
CREATE_SQL_SERVER_AUTHENTICATION_USERLOGIN {
    Query="CREATE LOGIN {__NAME__}, DEFAULT_LANGUAGE={defaultLanguage}
    WITH PASSWORD={__PASSWORD__}"
    QueryType="SQL"
    Parameters=["__NAME__": "Type:String",
    "defaultLanguage": "Type:String", "__PASSWORD__": "Type:GuardedString",
    Tags:QUOTES"]
    ExtensionJoin=","
    ExtensionSeparator=","
    QueryExtensions=["DEFAULT_DATABASE", "DEFAULT_LANGUAGE"]
}
```

- c. Save and close the query file.

7. Create a new bundle JAR file that contains the updated manifest file and the query files as follows:

- a. Open the command prompt and navigate to the temporary directory:

```
c:\temp
```

- b. If there are any driver JAR files, copy them to the lib directory under the bundle directory.

- c. Run the following command:

```
jar -cvfm org.identityconnectors.dbum-1.0.1117.jar META-INF/MANIFEST.MF *
```

The new connector bundle JAR name contains the new bundle version.

8. In the case of a remote connector server, copy the new bundle JAR file in the bundles directory of the remote connector server, instead of posting the JAR file to the Oracle Identity Manager database. Skip to Step 9.
9. Run the Oracle Identity Manager Update JARs utility to post the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

```
OIM_HOME/server/bin/UpdateJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UpdateJars.sh
```

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

10. Update the configuration lookup with the new bundle version.

For example, you can update the `Lookup.DBUM.MSSQL.Configuration` lookup definition.

3.7.2.2 Configuring Oracle Identity Manager

You can skip this procedure if the parameter you added already exists as a default form field in Oracle Identity Manager.

To configure Oracle Identity Manager for adding a parameter:

1. Log into Oracle Identity Manager Design Console.
2. Create a new version of the process form:
 - a. Expand **Development Tools**.
 - b. Double-click **Form Designer**.
 - c. Search for and open the **UD_DB_SQL_U** process form.

- d. Click **Create New Version**.
 On the Create a new version dialog box, enter a new version in the Label field, and then click the save icon.
3. Add the new field on the process form.
 - a. Click **Add**.
 A field is added to the list. Enter the details of the field.
 For example, if you are adding the CustomAttribute1 field, enter UD_DB_SQL_U_CUSTOM1 in the **Name** field and then enter the rest of the details of this field.
 - b. Click the save icon and then click **Make Version Active**.
4. If you are using Oracle Identity Manager release 11.1.2.x or later, then all changes made to the Form Designer of the Design Console must be done in a new UI form as follows:
 - a. Log in to Oracle Identity System Administration.
 - b. Create and active a sandbox.
 - c. Create a new UI form to view the newly added field along with the rest of the fields. See *Creating Forms By Using the Form Designer in Oracle Fusion Middleware Administering Oracle Identity Manager*.
 - d. Associate the newly created UI form with the application instance of your target system. To do so, open the existing application instance for your resource, from the Form field, select the form (created in Step 4.c), and then save the application instance.
 - e. Publish the sandbox as described in *Publishing a Sandbox of Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.
5. Create an entry for the field in the lookup definition for provisioning as follows:
 - a. Expand **Administration**.
 - b. Double-click **Lookup Definition**.
 - c. Search for and open the **Lookup.DBUM.MSSQL.UM.ProvAttrMap** lookup definition.
 - d. Click **Add** and enter the Code Key and Decode values for the field.
 The Code Key value must be the form field name. The Decode value must be the attribute name on the target system.
 For example, enter `Custom Attribute 1` in the **Code Key** field and then enter `CustomAttribute1` in the **Decode** field.
 - e. Click the save icon.
6. Create a process task to update the new field Custom Attribute 1 as follows:
 - a. Expand **Process Management**.
 - b. Double-click **Process Definition** and open the **MSSQL DB User** process definition.
 - c. Click **Add** and enter the task name, for example, `Custom Attribute 1 Updated`, and the task description.

- d. In the Task Properties section, select the **Conditional** and **Allow Multiple Instances** checkboxes and click the save icon.
- e. On the Integration tab, click **Add**, and then click **Adapter**.
- f. Select the **adpMSSQLUPDATEUSER** adapter, click the save icon, and then click **OK** in the message that is displayed.
- g. To map the adapter variables listed in this table, select the adapter, click **Map**, and then specify the data given in the following table:

Variable Name	Data Type	Map To	Qualifier	Literal Value
Adapter return value	Object	Response code	NA	NA
attributeName	String	Literal	String	Custom Attribute 1
itRes	String	Literal	String	UD_DB_SQL_U_ITRES
objectType	String	Literal	String	User
processInstanceKey	Long	Process Data	Process Instance	NA

- h. On the Responses tab, click **Add** to add the following response codes:

Code Name	Description	Status
ERROR	Error occurred	R
UNKNOWN	An unknown response was received	R
SUCCESS	Operation completed	C

- i. Click the save icon and then close the dialog box.

3.7.3 About Configuring the Connector for Multiple Installations of MSSQL

You might want to configure the connector for multiple installations of the target system. The following example illustrates this requirement:

The London and New York offices of Example Multinational Inc. have their own installations of the target system. The company has recently installed Oracle Identity Manager, and they want to configure Oracle Identity Manager to link all the installations of the target system.

To meet the requirement posed by such a scenario, you can create copies of connector objects, such as the IT resource and resource object.

The decision to create a copy of a connector object might be based on a requirement. For example, an IT resource can hold connection information for one target system installation. Therefore, it is mandatory to create a copy of the IT resource for each target system installation.

With some other connector objects, you do not need to create copies at all. For example, a single attribute-mapping lookup definition can be used for all installations of the target system.

All connector objects are linked. For example, a scheduled job holds the name of the IT resource. Similarly, the IT resource for a target system holds the name of the configuration lookup definition, Lookup.DBUM.MSSQL.Configuration. If you create a copy of an object, then you must specify the name of the copy in associated connector objects.

 **Note:**

- To reconcile data from a particular target system installation, specify the name of the IT resource for that target system installation as the value of the scheduled job attribute that holds the IT resource name. For example, you enter the name of the IT resource as the value of the IT resource attribute of the scheduled job that you run.
- When you use Identity Self Service to perform provisioning, you can specify the IT resource corresponding to the target system installation to which you want to provision the user.

Table 3-23 lists associations between connector objects whose copies can be created and the other objects that reference these objects. When you create a copy of a connector object, use this information to change the associations of that object with other objects.

 **Note:**

- On a particular Oracle Identity Manager installation, if you create a copy of a connector object, then you must set a unique name for it.
- If you are using Oracle Identity Manager release 11.1.2.x or later, then in addition to the procedure described in this section, you must create an application instance for each IT resource. See [Configuring Oracle Identity Manager Release 11.1.2 or Later](#) for information on creating an application instance.

Table 3-23 Connector Objects and Their Associations

Connector Object	Name	Referenced By	Comments on Creating a Copy
IT resource	MSSQL DB	Process forms: <ul style="list-style-type: none"> • UD_DB_SQL_L • UD_DB_SQL_U Scheduled Jobs	You need to create a copy of IT resource with a different name.
Resource object	<ul style="list-style-type: none"> • MSSQL DB User Login • MSSQL DB User 	All connector operations	It is optional to create a copy of the resource object. If you are reconciling the same set of attributes from all installations of the target system, then you need not create a copy of the resource object. Note: Create copies of the resource object only if there are differences in attributes between the various installations of the target system.

Table 3-23 (Cont.) Connector Objects and Their Associations

Connector Object	Name	Referenced By	Comments on Creating a Copy
Scheduled Jobs	There are many scheduled jobs for different purposes.	NA	You can use the scheduled jobs with the same names. However, you must update the values of the parameters depending on the target system you want to use.
Process definition	<ul style="list-style-type: none"> MSSQL DB User Login MSSQL DB User 	NA	<p>It is optional to create a copy of the process definition. If you are reconciling or provisioning the same set of attributes from all installations of the target system, then you need not create a copy of the process definition.</p> <p>Note: Create copies of the process form only if there are differences in attributes between the various installations of the target system.</p>
Process form	<ul style="list-style-type: none"> UD_DB_SQL_L UD_DB_SQL_U 	<p>Process definitions:</p> <ul style="list-style-type: none"> MSSQL DB User Login MSSQL DB User 	<p>It is optional to create a copy of the process form. If you are provisioning the same set of attributes from all installations of the target system, then you need not create a copy of the process definition.</p> <p>Note: Create copies of the process form only if there are differences in attributes between the various installations of the target system.</p>
Child process form	UD_DB_SQL_R	<ul style="list-style-type: none"> MSSQL DB User (Process definition) UD_DB_SQL_U (Process form) 	<p>It is optional to create a copy of the child process form. If you are provisioning a new set of child data, then you need to create a copy of the child and parent process form. Then, assign the newly created child process form to the newly created parent process form.</p>
Configuration lookup definition for a target system configured as a target resource	Lookup.DBUM.MSSQL.Configurat	MSSQL DB (IT resource)	<p>It is optional to create a copy of the configuration lookup definition. If you are provisioning and reconciling the same set of attributes in all installations of the target system (configured as a target resource), then you need not create a copy of the configuration lookup definition.</p> <p>Note: Create copies of the configuration lookup definition only if there are differences in attributes between the various installations of the target system and you have created a new process form.</p>

Table 3-23 (Cont.) Connector Objects and Their Associations

Connector Object	Name	Referenced By	Comments on Creating a Copy
Configuration lookup definition for a target system configured as a trusted source	Lookup.DBUM.M SSQL.Configuration.Trusted	MSSQL DB (IT resource)	<p>It is optional to create a copy of the configuration lookup definition. If you are reconciling the same set of attributes in all installations of the target system (configured as a trusted source), then you need not create a copy of the configuration lookup definition.</p> <p>Note: Create copies of the configuration lookup definition for trusted source only if there are differences in attributes between the various installations of the target system and you have created a new process form.</p>
Resource object attributes mapping lookup definition (for trusted source)	Lookup.DBUM.M SSQL.UM.Recon AttrMap.Trusted		<p>It is optional to create a copy of resource object attribute mapping lookup. If you are reconciling the same set of attributes in all installations of the target system (configured as a trusted source), then you need not to create a copy of resource object attribute mapping lookup.</p> <p>Note: Create copies of this lookup definition only if there are differences in attributes between the two installations of the target system.</p>

3.7.4 About Configuring the Connector for Multiple Trusted Source Reconciliation from MSSQL



Note:

This connector supports multiple trusted source reconciliation.

This section describes an optional procedure. Perform this procedure only if you want to configure the connector for multiple trusted source reconciliation.

The following are examples of scenarios in which there is more than one trusted source for user data in an organization:

- One of the target systems is a trusted source for data about users. The second target system is a trusted source for data about contractors. The third target system is a trusted source for data about interns.
- One target system holds the data of some of the identity fields that constitute an OIM User. Two other systems hold data for the remaining identity fields. In other

words, to create an OIM User, data from all three systems would need to be reconciled.

If the operating environment of your organization is similar to that described in either one of these scenarios, then this connector enables you to use the target system as one of the trusted sources of person data in your organization.

The following are the guidelines for configuring multiple trusted source reconciliation:

- By default, trusted source reconciliation is performed for MSSQL login entities.
To configure trusted source reconciliation to be performed for user entities, modify the **Resource Object Name** parameter of the reconciliation scheduled jobs to `MSSQL_DB User` and **Object Type** from `USERLOGIN` to `User`.
- To enable reconciliation for each new trusted source, create a new IT Resource and update the **IT Resource Name** parameter of the scheduled jobs with the name of the new IT Resource.

3.7.5 Configuring Validation of Data During Reconciliation and Provisioning for MSSQL

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

To configure validation of data:

1. Write code that implements the required validation logic in a Java class with a fully qualified domain name (FQDN), such as `org.identityconnectors.dbum.extension.DBUMValidator`.

This validation class must implement the `validate` method. The following sample validation class checks if the value in the First Name attribute contains the number sign (#):

```
package com.validationexample;

import java.util.HashMap;

public class MyValidator {
    public boolean validate(HashMap hmUserDetails, HashMap hmEntitlementDetails,
        String sField) throws ConnectorException {

        /* You must write code to validate attributes. Parent
           * data values can be fetched by using hmUserDetails.get(field)
           * For child data values, loop through the
           * ArrayList/Vector fetched by hmEntitlementDetails.get("Child
Table")

           * Depending on the outcome of the validation operation,
           * the code must return true or false.
           */

        /*
           * In this sample code, the value "false" is returned if the field
           * contains the number sign (#). Otherwise, the value "true" is
           * returned.
           */
        boolean valid = true;
        String sFirstName = (String) hmUserDetails.get(sField);
```

```

        for (int i = 0; i < sFirstName.length(); i++) {
            if (sFirstName.charAt(i) == '#') {
                valid = false;
                break;
            }
        }
        return valid;
    }
}

```

2. Log in to the Design Console.
3. Search for and open one of the lookup definitions (or create a new lookup) listed in [Lookup Definition for Validation of Data in MSSQL](#).

For example, **Lookup.DBUM.MSSQL.UM.ProvValidations**.

4. In the **Code Key** column, enter the resource object field name that you want to validate. For example, `Login Name`.
5. In the **Decode** column, enter the class name. For example, `org.identityconnectors.dbum.extension.DBUMValidator`.
6. Save the changes to the lookup definition.
7. Search for and open the configuration lookup definition for the target system you use.

For example, **Lookup.DBUM.MSSQL.UM.Configuration**.

8. In the **Code Key** column, enter one of the following entries:

- To configure validation of data for reconciliation:

`Recon Validation Lookup`

- To configure validation of data for provisioning:

`Provisioning Validation Lookup`

9. In the **Decode** column, enter the name of the lookup you updated or created in step 3.

For example, **Lookup.DBUM.MSSQL.UM.ProvValidations**.

For login entity, use **Lookup.DBUM.MSSQL.Login.ProvValidations**.

10. Save the changes to the lookup definition.
11. Create a JAR with the class and upload it to the Oracle Identity Manager database as follows:

Run the Oracle Identity Manager Upload JARs utility to post the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

OIM_HOME/server/bin/UploadJars.bat

For UNIX:

OIM_HOME/server/bin/UploadJars.sh

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

12. Run the PurgeCache utility to clear content related to request datasets from the server cache.
13. Perform reconciliation or provisioning to verify validation for the field, for example, Login Name.

3.7.6 Configuring Transformation of Data During User Reconciliation for MSSQL

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

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

1. Write code that implements the required transformation logic in a Java class with a fully qualified domain name (FQDN), such as `org.identityconnectors.dbum.extension.DBUMTransformation`.

This transformation class must implement the transform method. The following sample transformation class creates a value for the Full Name attribute by using values fetched from the First Name and Last Name attributes of the target system:

```
package com.transformationexample;

import java.util.HashMap;

public class MyTransformer {
    public Object transform(HashMap hmUserDetails, HashMap hmEntitlementDetails,
String sField) throws ConnectorException {
        /*
         * You must write code to transform the attributes.
         * Parent data attribute values can be fetched by
         * using hmUserDetails.get("Field Name").
         * To fetch child data values, loop through the
         * ArrayList/Vector fetched by hmEntitlementDetails.get("Child
Table")
         * Return the transformed attribute.
         */
        String sFirstName = (String) hmUserDetails.get("First Name");
        String sLastName = (String) hmUserDetails.get("Last Name");
        return sFirstName + "." + sLastName;
    }
}
```

2. Log in to the Design Console.

3. Search for and open one of the lookup definitions (or create a new lookup) listed in [Lookup Definitions for Transformation of Data in MSSQL](#).

For example, **Lookup.DBUM.MSSQL.UM.ReconTransformations**.

 **Note:**

If you cannot find these lookup definitions, create new lookup definitions.

4. In the **Code Key** column, enter the resource object field name you want to transform. For example, `Login Name`.
5. In the **Decode** column, enter the class name. For example, `org.identityconnectors.dbum.extension.DBUMTransformation`.
6. Save the changes to the lookup definition.
7. Search for and open the **Lookup.DBUM.MSSQL.UM.Configuration** lookup definition.
8. In the **Code Key** column, enter `Recon Transformation Lookup`.
9. In the **Decode** column, enter the name of the lookup you updated or created in step 3.

For example, **Lookup.DBUM.MSSQL.UM.ReconTransformations**.

For login entity, use **Lookup.DBUM.MSSQL.Login.ReconTransformation**.

10. Save the changes to the lookup definition.
11. Create a JAR with the class and upload it to the Oracle Identity Manager database as follows:

Run the Oracle Identity Manager Upload JARs utility to post the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

```
OIM_HOME/server/bin/UploadJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UploadJars.sh
```

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

12. Run the PurgeCache utility to clear content related to request datasets from the server cache.

13. Perform reconciliation to verify transformation of the field, for example, Login Name.

3.7.7 Configuring Resource Exclusion Lists for MSSQL

You can specify a list of accounts that must be excluded from reconciliation and provisioning operations. Accounts whose user IDs you specify in the exclusion list are not affected by reconciliation and provisioning operations.

In one of the lookup definitions for exclusion lists, enter the user IDs of target system accounts for which you do not want to perform provisioning and reconciliation operations. See [Lookup Definitions for Exclusion Lists for MSSQL](#) for information about the lookup definitions and the format of the entries in these lookups.

To add entries in the lookup for exclusions during provisioning operations for Oracle Database:

 **Note:**

To specify user IDs to be excluded during reconciliation operations, add entries in the `Lookup.DBUM.MSSQL.UM.ReconExclusions` lookup definition.

1. On the Design Console, expand **Administration** and then double-click **Lookup Definition**.
2. Search for and open the **Lookup.DBUM.MSSQL.UM.ProvExclusions** or **Lookup.DBUM.MSSQL.UM.ReconExclusions** lookup definition depending on provisioning or reconciliation exclusion lists.

For login entity, use **Lookup.DBUM.MSSQL.Login.ProvExclusions** or **Lookup.DBUM.MSSQL.Login.ReconExclusions**.

For trusted source reconciliation, use **Lookup.DBUM.MSSQL.UM.ExclusionList.Trusted**.

3. Click **Add**.
4. In the Code Key column, enter the resource object field name on which the exclusion list is applied. In the Decode column, enter the corresponding ID of the record to exclude.

For example, if you do not want to provision users with the user ID User001, then you must populate the lookup definition with the following values:

Code Key	Decode
Login Name	User001

 **Note:**

If you want to specify a list of accounts that must be excluded during reconciliation or provisioning, the code key value being specified here must be exactly as the corresponding code key value in the `Lookup.DBUM.MSSQL.UM.ReconAttrMap` lookup definition, or in the `Lookup.DBUM.MSSQL.UM.ProvAttrMap` lookup definition, respectively.

- If there is more than one user ID to exclude, then in the decode column, enter a list of all user IDs to exclude. Note that each User ID must be separated by a vertical bar (|).

For example, if you do not want to provision users with user IDs User001, User002, and User088 then you must populate the lookup definition with the following values:

Code Key	Decode
Login Name	User001 User002 User088

You can also perform pattern matching to exclude user accounts. You can specify regular expressions supported by the representation in the `java.util.regex.Pattern` class.

See Also:

For information about the supported patterns, visit <http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html>

For example, if you do not want to provision users matching any of the user IDs User001, User002, and User088, then you must populate the lookup definition with the following values:

Code Key	Decode
Login Name[PATTERN]	User001 User002 User088

If you do not want to provision users whose user IDs start with 00012, then you must populate the lookup definition with the following values:

Code Key	Decode
Login Name[PATTERN]	00012*

- Click the save icon.

3.7.8 Setting Up Action Scripts for MSSQL

Learn about action scripts and how to configure them to run before or after the create, update, or delete an account provisioning operations.

This section provides information about the following topics:

- [About Action Scripts for MSSQL](#)
- [Configuring Action Scripts for MSSQL](#)

3.7.8.1 About Action Scripts for MSSQL

Actions are scripts that you can configure to run before or after the create, update, or delete an account provisioning operations. For example, you could configure a script

to run before every user creation. In another scenario, suppose you have a table called `AUDIT_USERLOG` where you want to log user creation activities performed only by the connector. Then, you could create and use after create script for adding data to this table after create operation.

**Note:**

To configure a before or after action, your connector must support running scripts. An exception is Groovy (with target set to **Connector**), which the Identity Connector Framework (ICF) supports by default for all converged connectors.

Every connector should specify which scripting language and which target it supports. This connector supports the following script:

CMD: windows batch script and **target:** Connector

The target refers to the location where the script is executed. In this case, the script is executed on the same computer (JVM or .NET Runtime) where the connector is deployed. For example, if you deploy the connector on the connector server, the script will be executed on that computer.

That is, if you are using a local framework, the script runs in your JVM. If you are connected to a remote framework, the script runs in the remote JVM or .NET Runtime.

3.7.8.2 Configuring Action Scripts for MSSQL

To configure the action:

1. Log in to the Design Console.
2. Search for and open the **Lookup.DBUM.MSSQL.UM.Configuration** lookup definition.

For user login entity, search for and open the **Lookup.DBUM.MSSQL.Login.Configuration** lookup definition.

3. Add the following new values:
 - **Code Key:** Before Create Action Language
 - **Decode:** Enter the scripting language of the script you want to execute
 - Sample values: `SQL` or `STOREDPROC`
4. Add these new values:
 - **Code Key:** Before Create Action File
 - **Decode:** Enter the full path to the file containing the script to be executed (Oracle Identity Manager must be able to access this file.)
 - **Example:** `/home/scripts/testscript.sql`

This script may have a query as follows:

```
INSERT INTO AUDIT_USERLOG VALUES ({{__NAME__}}, CURRENT_TIMESTAMP))
```

5. Add these new values:
 - **Code Key:** Before Create Action Target
 - **Decode:** Connector

As previously stated, the connector supports the CMD script for a Connector target.

6. Save the lookup definition.

Now, this action will be executed every time you create a user. You must configure these three values for each action you want to execute.

4

Using and Extending the Connector for Oracle Database

You can use the Database User Management Connector for Oracle Database for performing reconciliation and provisioning operations after configuring it to meet your requirements. You can also extend the functionality of the connector to address your specific business requirements.

This chapter contains the following topics:

Note:

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

For Oracle Identity Manager hosted on a Microsoft Windows computer, if you have a previously installed connector, then you must extract the connector bundle zip file again before installing a new connector.

Database drivers are not needed as they are already loaded for Oracle Identity Manager operations. However, if you want to use the connector with previous versions of database (such as Oracle 9i), then you must use a remote connector server.

- [Configuring Secure Communication Between Oracle Database and Oracle Identity Manager](#)
- [Determining Values for the JDBC URL and Connection Properties Parameters for Oracle Database](#)
- [Lookup Definitions Used During Connector Operations for Oracle Database](#)
- [Scheduled Jobs for Oracle Database](#)
- [Reconciliation from Oracle Database](#)
- [Provisioning for Oracle Database](#)
- [Extending the Connector for Oracle Database](#)

4.1 Configuring Secure Communication Between Oracle Database and Oracle Identity Manager



Note:

It is recommended that you perform the procedure described in this section to secure communication between the target system and Oracle Identity Manager.

To secure communication between Oracle Database and Oracle Identity Manager, you can perform either one or both of the following procedures:

- [Configuring Data Encryption and Integrity in Oracle Database](#)
- [Configuring SSL Communication in Oracle Database](#)

4.1.1 Configuring Data Encryption and Integrity in Oracle Database

To configure data encryption and integrity, see Data Encryption in *Oracle Database Advanced Security Administrator's Guide*.

4.1.2 Configuring SSL Communication in Oracle Database

To enable SSL communication between Oracle Database and Oracle Identity Manager:

1. See *Oracle Database Advanced Security Administrator's Guide* for information about enabling SSL communication between Oracle Database and Oracle Identity Manager.

Export the certificate on the Oracle Database host computer.

2. Copy the certificate to Oracle Identity Manager.
3. Import the certificate into the JVM truststore of the application server on which Oracle Identity Manager is running.

To import the certificate into the truststore, run the following command:

```
..\..\bin\keytool -import -file FILE_LOCATION -keystore TRUSTSTORE_LOCATION -storepass TRUSTSTORE_PASSWORD -trustcacerts -alias ALIAS
```

In this command:

- Replace *FILE_LOCATION* with the full path and name of the certificate file.
- Replace *ALIAS* with an alias for the certificate.
- Replace *TRUSTSTORE_PASSWORD* with a password for the truststore.
- Replace *TRUSTSTORE_LOCATION* with one of the truststore paths from [Table 4-1](#). This table shows the location of the truststore for each of the supported application servers.

 **Note:**

In an Oracle Identity Manager cluster, import the file into the truststore on each node of the cluster.

Table 4-1 Truststore Locations on Supported Application Servers

Application Server	Truststore Location
Oracle WebLogic Server	<ul style="list-style-type: none"> If you are using Oracle jrockit_R27.3.1-jdk, then import the certificate into the keystore in the following directory: <i>JROCKIT_HOME/jre/lib/security</i> If you are using the default Oracle WebLogic Server JDK, then import the certificate into the keystore in following directory: <i>WEBLOGIC_HOME/java/jre/lib/security/cacerts</i> If you are using a JDK other than Oracle jrockit_R27.3.1-jdk or Oracle WebLogic Server JDK, then import the certificate into your keystore at the following directory: <i>JAVA_HOME/jre/lib/security/cacerts</i>

- To enable secure communication between Oracle Database and Oracle Identity Manager, set the value of the UseSSL IT resource parameter to `true`. You must provide a value for this parameter while performing the procedure described in [Configuring the IT Resource for the Connector Server](#).

4.2 Determining Values for the JDBC URL and Connection Properties Parameters for Oracle Database

This section discusses the JDBC URL and Connection Properties parameters. You apply the information in this section while performing the procedure described in [Configuring the IT Resource for the Target System](#).

The values that you specify for the JDBC URL and Connection Properties parameters depend on the security measures that you have implemented:

- [Only Data Encryption and Integrity Is Configured](#)
- [Only SSL Communication Is Configured](#)
- [Both Data Encryption and Integrity and SSL Communication Are Configured](#)

4.2.1 Only Data Encryption and Integrity Is Configured

If you have configured only data encryption and integrity, then enter the following values:

- JDBC URL parameter**

While configuring the IT resource, the value that you specify for the JDBC URL parameter must be in the following format:

```
jdbc:oracle:thin:@TARGET_HOST_NAME_or_IP_ADDRESS:PORT_NUM:sid
```

The following is a sample value for the JDBC URL parameter:

```
jdbc:oracle:thin:@ten.mydomain.com:1521:cust_db
```

- **Connection Properties parameter**

After you configure data encryption and integrity, the connection properties are recorded in the `sqlnet.ora` file. The value that you must specify for the Connection Properties parameter is explained by the following sample scenario:

 **See Also:**

Oracle Database Advanced Security Administrator's Guide for information about the `sqlnet.ora` file

Suppose the following entries are recorded in the `sqlnet.ora` file:

```
SQLNET.ENCRYPTION_SERVER=REQUIRED
SQLNET.ENCRYPTION_TYPES_SERVER=(3DES168, DES40, DES, 3DES112)
SQLNET.CRYPTO_CHECKSUM_SERVER=REQUESTED
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER=(SHA1,MD5)
```

While configuring the IT resource, you must specify the following as the value of the Connection Properties parameter:

 **Note:**

- The property-value pairs must be separated by number signs (#).
- As shown in the following example, for the `encryption_types` and `crypto_checksum_types` properties, you can select any of the values recorded in the `sqlnet.ora` file.

```
oracle.net.encryption_client=REQUIRED,oracle.net.encryption_types_client=(3DES168),oracle.net.crypto_checksum_client=REQUESTED,oracle.net.crypto_checksum_types_client=(MD5)
```

4.2.2 Only SSL Communication Is Configured

After you configure SSL communication, the JDBC URL is recorded in the `tnsnames.ora` file. See *Oracle Database Net Services Reference* for detailed information about the `tnsnames.ora` file.

The following are sample formats of the contents of the `tnsnames.ora` file. In these formats, `DESCRIPTION` contains the connection descriptor, `ADDRESS` contains the protocol address, `CONNECT_DATA` contains the database service identification information, and `SECURITY` contains SSL-specific information.

Sample Format 1:

```
NET_SERVICE_NAME=
(DESCRIPTION=
  (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
  (CONNECT_DATA=
    (SERVICE_NAME=SERVICE_NAME))
  (SECURITY_DN=(SSL_SERVER_CERT_DN="CN=server_test,C=US")))
```

Sample Format 2:

```

NET_SERVICE_NAME=
  (DESCRIPTION_LIST=
    (DESCRIPTION=
      (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
      (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
      (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
      (CONNECT_DATA=
        (SERVICE_NAME=SERVICE_NAME))
        (SECURITY_DN=(SSL_SERVER_CERT_DN="CN=server_test,C=US"))
      (DESCRIPTION=
        (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
        (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
        (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
        (CONNECT_DATA=
          (SERVICE_NAME=SERVICE_NAME))))))

```

Sample Format 3:

```

NET_SERVICE_NAME=
  (DESCRIPTION=
    (ADDRESS_LIST=
      (LOAD_BALANCE=on)
      (FAILOVER=off)
      (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
      (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION)))
    (ADDRESS_LIST=
      (LOAD_BALANCE=off)
      (FAILOVER=on)
      (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
      (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION)))
    (CONNECT_DATA=
      (SERVICE_NAME=SERVICE_NAME))
      (SECURITY_DN=(SSL_SERVER_CERT_DN="CN=server_test,C=US"))

```

If you have configured only SSL communication and imported the certificate that you create on the target system host computer into the JVM truststore of Oracle Identity Manager, then enter the following values:

JDBC URL parameter

While configuring the IT resource, the value that you specify for the JDBC URL parameter must be derived from the value of *NET_SERVICE_NAME* in the *tnsnames.ora* file. For example:

 **Note:**

As shown in this example, you must include only the `(ADDRESS=(PROTOCOL=TCPS) (HOST=HOST_NAME) (PORT=2484))` element because you are configuring SSL. You need not include other `(ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))` elements.

```

jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCPS) (HOST=myhost)
(PORT=2484))) (CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=mysid))
(SECURITY_DN=(SSL_SERVER_CERT_DN="CN=server_test,C=US"))

```

Connection Properties parameter

Whether you need to specify a value for the Connection Properties parameter depends on the truststore into which you import the certificate:

- If you import the certificate into the truststore of the JVM that Oracle Identity Manager is using, then you need not specify a value for the Connection Properties parameter.
- If you import the certificate into any other truststore, then while creating the connector, specify a value for the Connection Properties parameter in the following format:

```
javax.net.ssl.trustStore=STORE_LOCATION#javax.net.ssl.trustStoreType=JKS#java
x.net.ssl.trustStorePassword=STORE_PASSWORD
```

When you specify this value, replace *STORE_LOCATION* with the full path and name of the truststore, and replace *STORE_PASSWORD* with the password of the truststore.

4.2.3 Both Data Encryption and Integrity and SSL Communication Are Configured

If both data encryption and integrity and SSL communication are configured, then:

- **JDBC URL parameter**

While configuring the IT resource, to specify a value for the JDBC URL parameter, enter a comma-separated combination of the values for the JDBC URL parameter described in [Only Data Encryption and Integrity Is Configured](#) and [Only SSL Communication Is Configured](#).

Note:

The following examples represent Oracle RAC URL configurations.

For example:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCPS)
(HOST=myhost)(PORT=2484))) (CONNECT_DATA=(SERVER=DEDICATED)
(SERVICE_NAME=mysid)))
```

The following are guidelines on specifying the JDBC URL and Connection Properties parameters:

While configuring the IT resource, the value that you specify for the JDBC URL parameter must be in the following format:

Note:

The JDBC URL connection string must not exceed 200 characters.

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(HOST=HOST1_NAME.DOMAIN)(PORT=PORT1_NUMBER))(ADDRESS=(PROTOCOL=TCP)
(HOST=HOST2_NAME.DOMAIN)(PORT=PORT2_NUMBER))(ADDRESS=(PROTOCOL=TCP)
```

```
(HOST=HOST3_NAME.DOMAIN) (PORT=PORT3_NUMBER) . . . (ADDRESS=(PROTOCOL=TCP)
(HOST=HOSTn_NAME.DOMAIN) (PORT=PORTn_NUMBER)
(CONNECT_DATA=(SERVICE_NAME=ORACLE_DATABASE_SERVICE_NAME)))
```

Sample value:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST= host1.example.com)
(PORT=1521)) (ADDRESS=(PROTOCOL=TCP) (HOST= host2.example.com) (PORT=1521))
(ADDRESS=(PROTOCOL=TCP) (HOST= host3.example.com) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP)
(HOST= host4.example.com) (PORT=1521)) (CONNECT_DATA=(SERVICE_NAME= srvcel)))
```

- **Connection Properties parameter**

While configuring the IT resource, to specify a value for the Connection Properties parameter, enter a comma-separated combination of the values for the Connection Properties parameter described in [Only Data Encryption and Integrity Is Configured](#) and [Only SSL Communication Is Configured](#).

For example:

```
oracle.net.encryption_client=REQUIRED#oracle.net.encryption_types_client=(3DES168)#
oracle.net.crypto_checksum_client=REQUESTED,oracle.net.crypto_checksum_types_client
=(MD5)#javax.net.ssl.trustStore=STORE_LOCATION,javax.net.ssl.trustStoreType=JKS#jav
ax.net.ssl.trustStorePassword=STORE_PASSWORD
```

As shown in the following example, for the `encryption_types` and `crypto_checksum_types` properties, you can select any of the values recorded in the `sqlnet.ora` file. When you specify this value, replace `STORE_LOCATION` with the full path and name of the truststore, and replace `STORE_PASSWORD` with the password of the truststore.

4.3 Lookup Definitions Used During Connector Operations for Oracle Database

Lookup definitions used during connector operations can be categorized as follows:

- [Lookup Definitions Synchronized with Oracle Database](#)
- [Lookup Definitions for Configurations for Oracle Database](#)
- [Lookup Definitions for Attribute Mappings for Oracle Database](#)
- [Lookup Definitions for Exclusion Lists for Oracle Database](#)
- [Lookup Definitions for Transformation of Data in Oracle Database](#)
- [Lookup Definition for Validation of Data in Oracle Database](#)

4.3.1 Lookup Definitions Synchronized with Oracle Database

During a provisioning operation, you use a lookup field on the process form to specify a single value from a set of values. For example, you use the Role lookup field to select a role to be assigned to the user from the list of available roles. When you deploy the connector, lookup definitions corresponding to the lookup fields on the target system are created in Oracle Identity Manager. Lookup field synchronization involves copying additions or changes made to the target system lookup fields into the lookup definitions in Oracle Identity Manager.

The connector provides predefined SQL queries for fetching values from the target system lookup fields into the lookup definitions in Oracle Identity Manager. These predefined SQL queries are stored in the `LoVSearch.queries` file within the connector bundle.

After lookup definition synchronization, data is stored in the following format:

- Code Key value: *IT_RESOURCE_KEY~LOOKUP_FIELD_ID*

In this format:

- *IT_RESOURCE_KEY* is the numeric code assigned to each IT resource in Oracle Identity Manager.
- *LOOKUP_FIELD_ID* is the target system code assigned to each lookup field entry.

Sample value: 1~SYS_ADM

- Decode value: *IT_RESOURCE_NAME~LOOKUP_FIELD_ID*

In this format:

- *IT_RESOURCE_NAME* is the name of the IT resource in Oracle Identity Manager.
- *LOOKUP_FIELD_ID* is the target system code assigned to each lookup field entry.

Sample value: Oracle DB~SYS_ADM

While performing a provisioning operation in Identity Self Service, you select the IT resource for the target system on which you want to perform the operation. When you perform this action, the lookup definitions on the page are automatically populated with values corresponding to the IT resource (target system installation) that you select. If your environment has multiple installations of the target system, then values corresponding to all IT resources are displayed.

[Table 4-2](#) lists column names of the tables in Oracle Database that are synchronized with their corresponding lookup definitions in Oracle Identity Manager.

Table 4-2 Lookup Definitions Synchronized with Oracle Database

Lookup Definition	Target Table Name	Target Column Name
Lookup.DBUM.Oracle.AuthType	dba_users	DECODE(PASSWORD, 'EXTERNAL', 'EXTERNAL', 'GLOBAL', 'GLOBAL', 'PASSWORD')
Lookup.DBUM.Oracle.Privileges	DBA_SYS_PRIVS	PRIVILEGE
Lookup.DBUM.Oracle.Profiles	dba_users	DISTINCT profile
Lookup.DBUM.Oracle.Roles	DBA_ROLE_PRIVS	GRANTED_ROLE
Lookup.DBUM.Oracle.Temp.Tablespace	dba_users	DEFAULT_TABLESPACE
Lookup.DBUM.Oracle.Tablespaces	dba_users	TEMPORARY_TABLESPACE
Lookup.DBUM.Oracle.WithAdminOption	DBA_SYS_PRIVS, DBA_ROLE_PRIVS	ADMIN_OPTION

The Lookup.DBUM.Oracle.AuthType lookup definition holds information about authentication types that you can select for a target system account (login or user) that you create through Oracle Identity Manager.

Table 4-3 Entries in Lookup.DBUM.Oracle.AuthType

Code Key	Decode Key
EXTERNAL	EXTERNAL
GLOBAL	GLOBAL
PASSWORD	PASSWORD

4.3.2 Lookup Definitions for Configurations for Oracle Database

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

This section provides information about the following lookup definitions

- [Lookup.DBUM.Oracle.Configuration](#)
- [Lookup.DBUM.Oracle.UM.Configuration](#)
- [Lookup.DBUM.Oracle.Configuration.Trusted](#)
- [Lookup.DBUM.Oracle.UM.Configuration.Trusted](#)

4.3.2.1 Lookup.DBUM.Oracle.Configuration

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

Table 4-4 Entries in Lookup.DBUM.Oracle.Configuration

Code Key	Decode Key	Description
Bundle Name	org.identityconnectors.dbum	Name of the connector bundle package Do not modify this entry.
Bundle Version	1.0.1116	Version of the connector bundle class Do not modify this entry.
Connector Name	org.identityconnectors.dbum.DBUMConnector	Name of the connector class Do not modify this entry.
disableValuesSet	"EXPIRED & LOCKED", "LOCKED", "EXPIRED"	Possible values for the disabled status of a user

Table 4-4 (Cont.) Entries in Lookup.DBUM.Oracle.Configuration

Code Key	Decode Key	Description
reservedWordsList	"DROP", "INSERT", "ALTER", "CREATE", , "DELETE", "UPDATE", "GRANT", "TRUNC ATE", "EXEC", "TEMPORARY", "TABLESPACE", "DEFAULT", "QUOTA", "PROFILE", "IDENTIFIED", " EXTERNALLY", "AS", "GLOBALLY", "REVOKE", "ACCOUN T", "UNLOCK", "LOCK", "CASCADE"	Words that are not allowed to be used in attribute values that are used in the final SQL query of the connector operations
unsupportedChars	"&", "--", "~", "`", "\""	Characters that are not allowed to be used in attribute values that are used in the final SQL query of the connector operations
User Configuration Lookup	Lookup.DBUM.Oracle.UM.Configuration	Name of the lookup definition that contains user-specific configuration properties Do not modify this entry.

4.3.2.2 Lookup.DBUM.Oracle.UM.Configuration

The Lookup.DBUM.Oracle.UM.Configuration lookup definition holds user-specific connector configuration entries that are used during target resource reconciliation and provisioning operations.

Table 4-5 Entries in Lookup.DBUM.Oracle.UM.Configuration

Code Key	Decode Key
Provisioning Attribute Map	Lookup.DBUM.Oracle.UM.ProvAttrMap
Provisioning Exclusion List	Lookup.DBUM.Oracle.UM.ExclusionList
Provisioning Validation Lookup	Lookup.DBUM.Oracle.UM.ProvValidations
Recon Validation Lookup	Lookup.DBUM.Oracle.UM.ReconValidations
Recon Attribute Map	Lookup.DBUM.Oracle.UM.ReconAttrMap
Recon Exclusion List	Lookup.DBUM.Oracle.UM.ExclusionList
Recon Transformation Lookup	Lookup.DBUM.Oracle.UM.ReconTransformations

4.3.2.3 Lookup.DBUM.Oracle.Configuration.Trusted

The Lookup.DBUM.Oracle.Configuration.Trusted lookup definition holds connector configuration entries that are used during reconciliation and provisioning operations in trusted source mode.

Table 4-6 Entries in Lookup.DBUM.Oracle.Configuration.Trusted

Code Key	Decode Key
Bundle Name	org.identityconnectors.dbum
Bundle Version	1.0.1116
Connector Name	org.identityconnectors.dbum.DBUMConnector
disableValuesSet	"EXPIRED & LOCKED", "LOCKED", "EXPIRED"
User Configuration Lookup	Lookup.DBUM.Oracle.UM.Configuration.Trusted

4.3.2.4 Lookup.DBUM.Oracle.UM.Configuration.Trusted

The Lookup.DBUM.Oracle.UM.Configuration.Trusted lookup definition holds user-specific connector configuration entries that are used during reconciliation and provisioning operations in trusted source mode.

Table 4-7 Entries in Lookup.DBUM.Oracle.UM.Configuration.Trusted

Code Key	Decode Key
Recon Attribute Defaults	Lookup.DBUM.Oracle.UM.ReconDefaults.Trusted
Recon Attribute Map	Lookup.DBUM.Oracle.UM.ReconAttrMap.Trusted
Recon Validation Lookup	Lookup.DBUM.Oracle.UM.ReconValidations.Trusted
Recon Exclusion List	Lookup.DBUM.Oracle.UM.ExclusionList.Trusted
Recon Transformation Lookup	Lookup.DBUM.Oracle.UM.ReconTransformations.Trusted

4.3.3 Lookup Definitions for Attribute Mappings for Oracle Database

This section describes the following lookup definitions:

- [Lookup.DBUM.Oracle.UM.ProvAttrMap](#)
- [Lookup.DBUM.Oracle.UM.ReconAttrMap](#)
- [Lookup.DBUM.Oracle.UM.ReconAttrMap.Trusted](#)
- [Lookup.DBUM.Oracle.UM.ReconDefaults.Trusted](#)

4.3.3.1 Lookup.DBUM.Oracle.UM.ProvAttrMap

The Lookup.DBUM.Oracle.UM.ProvAttrMap lookup definition holds user-specific mappings between process form fields (Code Key values) and target system attributes (Decode values) used during provisioning operations.

Table 4-8 Entries in Lookup.DBUM.Oracle.UM.ProvAttrMap

Code Key	Decode Key
Authentication Type	authType
Default Tablespace[LOOKUP]	tablespace

Table 4-8 (Cont.) Entries in Lookup.DBUM.Oracle.UM.ProvAttrMap

Code Key	Decode Key
Default Tablespace Quota (in MB)	defaultQuota
Global DN	globalDN
Password	__PASSWORD__
Profile Name[LOOKUP]	profile
Return Id	__UID__
Temporary Tablespace[LOOKUP]	tempTableSpace
UD_DB_ORA_P~Privilege[LOOKUP]	privileges~DBPrivilege~__NAME__
UD_DB_ORA_P~Privilege Admin Option	privileges~DBPrivilege~adminOption
UD_DB_ORA_R~Role[LOOKUP]	roles~DBRole~__NAME__
UD_DB_ORA_R~Role Admin Option	roles~DBRole~adminOption
Username	__NAME__

4.3.3.2 Lookup.DBUM.Oracle.UM.ReconAttrMap

The Lookup.DBUM.Oracle.UM.ReconAttrMap lookup definition holds user-specific mappings between reconciliation attribute names as specified in the resource object (Code Key values) and target system attributes (Decode values) used during reconciliation operations.

Table 4-9 Entries in Lookup.DBUM.Oracle.UM.ReconAttrMap

Code Key	Decode Key
Account Status	status
Authentication Type	authType
Default Tablespace[LOOKUP]	tablespace
Default Tablespace Quota	defaultQuota
Global DN	globalDN
Privilege List~Privilege Admin Option	privileges~DBPrivilege~adminOption
Privilege List~Privilege Name[LOOKUP]	privileges~DBPrivilege~__NAME__
Profile Name[LOOKUP]	profile
Reference ID	__UID__
Role List~Role Admin Option	roles~DBRole~adminOption

Table 4-9 (Cont.) Entries in Lookup.DBUM.Oracle.UM.ReconAttrMap

Code Key	Decode Key
Role List~Role Name[LOOKUP]	roles~DBRole~__NAME__
Status	__ENABLE__
Temporary Tablespace[LOOKUP]	tempTableSpace
User Name	__UID__

4.3.3.3 Lookup.DBUM.Oracle.UM.ReconAttrMap.Trusted

The Lookup.DBUM.Oracle.UM.ReconAttrMap.Trusted lookup definition holds user-specific mappings between reconciliation attribute names as specified in the resource object (Code Key values) and target system attributes (Decode values) used during reconciliation operations in trusted source mode.

Table 4-10 Entries in Lookup.DBUM.Oracle.UM.ReconAttrMap.Trusted

Code Key	Decode Key
First Name	__UID__
Status[TRUSTED]	__ENABLE__
User ID	__UID__

4.3.3.4 Lookup.DBUM.Oracle.UM.ReconDefaults.Trusted

This lookup definition contains the default values for the Oracle Identity Manager user attributes. You can change these values as per your requirements.

For example, if you want the users reconciled from a trusted source to be part of the MyORG organization, then map the lookup definition entry as follows:

Code Key = Organization Name

Decode = MyORG (instead of Xellerate Users)

Table 4-11 Entries in Lookup.DBUM.Oracle.UM.ReconDefaults.Trusted

Code Key	Decode Key
Empl Type	Full-Time
Organization Name	Xellerate Users
Status	Active
User Type	End-User

4.3.4 Lookup Definitions for Exclusion Lists for Oracle Database

These lookup definitions hold resources for which you do not want to perform provisioning and reconciliation operations. Exclusions can be applied to any attribute in the process form or reconciliation profile. The Code Key value must be one of the Code Key values in Lookup.DBUM.Oracle.UM.ReconAttrMap or Lookup.DBUM.Oracle.UM.ProvAttrMap lookup definitions.

Depending on how the target system is configured, you can use one of the following lookups:

- For target resource mode: Lookup.DBUM.Oracle.UM.ExclusionList
- For trusted source mode: Lookup.DBUM.Oracle.UM.ExclusionList.Trusted

The following is the format of the values stored in these lookups:

Code Key	Decode	Sample Values
User Name	User ID of a user	Code Key: User Name Decode: User001
User Name with the [PATTERN] suffix	A regular expression supported by the representation in the <code>java.util.regex.Pattern</code> class	Code Key: User Name[PATTERN] To exclude users matching any of the user ID 's User001, User002, User088, then: Decode: User001 User002 User088 To exclude users whose user ID 's start with 00012, then: Decode: 00012* See Also: For information about the supported patterns, visit http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html

[Configuring Resource Exclusion Lists for Oracle Database](#) describes the procedure to add entries in these lookup definitions.

4.3.5 Lookup Definitions for Transformation of Data in Oracle Database

These lookup definitions hold resources for which you want to enable transformation of data during reconciliation operations.

Depending on how the target system is configured, use one of the following lookup definitions:

- For target resource mode: Lookup.DBUM.Oracle.UM.ReconTransformations

Table 4-12 Entries in Lookup.DBUM.Oracle.UM.ReconTransformations

Code Key	Decode Key
Privilege List	oracle.iam.connectors.dbum.transformations.OraclePrivilegeAdminOptionTransformation
Role List	oracle.iam.connectors.dbum.transformations.OracleRoleAdminOptionTransformation

- For trusted source mode: Lookup.DBUM.Oracle.UM.ReconTransformations.Trusted

[Configuring Transformation of Data During User Reconciliation for Oracle Database](#) describes the procedure to add entries in these lookup definitions.

4.3.6 Lookup Definition for Validation of Data in Oracle Database

You can use the Lookup.DBUM.Oracle.UM.ProvValidations lookup to configure validation of data during provisioning operations.

[Configuring Validation of Data During Reconciliation and Provisioning for Oracle Database](#) describes the procedure to add entries in this lookup definition.

4.4 Scheduled Jobs for Oracle Database

When you run the Connector Installer or import the connector XML file, the scheduled jobs are automatically created in Oracle Identity Manager.

This section describes the following topics:

- [Scheduled Jobs for Lookup Field Synchronization for Oracle Database](#)
- [Attributes of the Scheduled Jobs for Oracle Database](#)
- [Configuring Scheduled Jobs for Oracle Database](#)

4.4.1 Scheduled Jobs for Lookup Field Synchronization for Oracle Database

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

The following scheduled jobs are used for lookup field synchronization:

- DBUM Oracle Privileges Lookup Reconciliation
- DBUM Oracle Profile Lookup Reconciliation
- DBUM Oracle Roles Lookup Reconciliation
- DBUM Oracle Tablespaces Lookup Reconciliation
- DBUM Oracle Temporary Tablespaces Lookup Reconciliation

You must specify values for the attributes of these scheduled jobs. [Table 4-13](#) describes the attributes of these scheduled jobs.

Table 4-13 Attributes of the Scheduled Jobs for Lookup Field Synchronization

Attribute	Description
Code Key Attribute	<p>Enter the name of the connector or target system attribute that is used to populate the Code Key column of the lookup definition (specified as the value of the Lookup Name attribute).</p> <p>Sample value: <code>__NAME__</code></p> <p>Note: Do not change the value of this attribute.</p>
Decode Attribute	<p>Enter the name of the connector or target system attribute that is used to populate the Decode column of the lookup definition (specified as the value of the Lookup Name attribute).</p> <p>Sample value: <code>__NAME__</code></p>
IT Resource Name	<p>Enter the name of the IT resource for the target system installation from which you want to reconcile user records.</p> <p>Default value: <code>Oracle DB</code></p>
Lookup Name	<p>This attribute holds the name of the lookup definition that maps each lookup definition with the data source from which values must be fetched.</p> <p>Depending on the scheduled job you are using, the default values are as follows:</p> <ul style="list-style-type: none"> • For DBUM Oracle Privileges Lookup Reconciliation - <code>Lookup.DBUM.Oracle.Privileges</code> • For DBUM Oracle Profile Lookup Reconciliation - <code>Lookup.DBUM.Oracle.Profiles</code> • For DBUM Oracle Roles Lookup Reconciliation - <code>Lookup.DBUM.Oracle.Roles</code> • For DBUM Oracle Tablespaces Lookup Reconciliation - <code>Lookup.DBUM.Oracle.Tablespaces</code> • For DBUM Oracle Temporary Tablespaces Lookup Reconciliation - <code>Lookup.DBUM.Oracle.Temp.Tablespace</code>
Object Type	<p>Enter the type of object whose values must be synchronized.</p> <p>Depending on the scheduled job you are using, the default values are as follows:</p> <ul style="list-style-type: none"> • For DBUM Oracle Privileges Lookup Reconciliation - <code>__PRIVILEGES__</code> • For DBUM Oracle Profile Lookup Reconciliation - <code>__PROFILE__</code> • For DBUM Oracle Roles Lookup Reconciliation - <code>__ROLES__</code> • For DBUM Oracle Tablespaces Lookup Reconciliation - <code>__TABLESPACES__</code> • For DBUM Oracle Temporary Tablespaces Lookup Reconciliation - <code>__TEMPTABLESPACES__</code> <p>Note: Do not change the value of this attribute.</p>
Resource Object Name	<p>Enter the name of the resource object that is used for reconciliation.</p> <p>Default value: <code>Oracle DB User</code></p>

4.4.2 Attributes of the Scheduled Jobs for Oracle Database

The following scheduled jobs are used to reconcile user data in the target resource (account management) mode of the connector:

- DBUM Oracle User Target Reconciliation
- DBUM Oracle Delete User Target Reconciliation

The following scheduled jobs are used to reconcile user data in the trusted source (identity management) mode of the connector:

- DBUM Oracle User Trusted Reconciliation
- DBUM Oracle Delete User Trusted Reconciliation

Table 4-14 describes the attributes of the scheduled jobs for user operations.

Table 4-14 Attributes of the Scheduled Jobs for Reconciliation

Attribute	Description
Batch Size	Value for running the scheduled job in batch mode. By default, this value is empty.
Filter	Expression for filtering records that must be reconciled by the scheduled job By default, the value of this attribute is empty. Sample value: <code>equalTo('__UID__', 'SEPT12USER1')</code> See Performing Limited Reconciliation from Oracle Database for the syntax of this expression.
Incremental Recon Attribute	Time stamp at which the last reconciliation run started Sample value: <code>lastModified</code> Note: Do not enter a value for this attribute. The reconciliation engine automatically enters a value for this attribute.
IT Resource Name	Name of the IT resource for the target system installation from which you want to reconcile user records Default value: <code>Oracle DB</code>
Latest Token	This attribute is used for internal purposes. By default, this value is empty.
Object Type	Type of object you want to reconcile Default value: <code>User</code>
Resource Object Name	Name of the resource object that is used for reconciliation Default value: <code>Oracle DB User</code>
Scheduled Task Name	Name of the scheduled job Note: For the scheduled job included with this connector, you must not change the value of this attribute. However, if you create a copy of the task, then you can enter the unique name for that scheduled job as the value of this attribute.

Table 4-15 describes the attributes of the scheduled jobs for delete operations.

Table 4-15 Attributes of the Scheduled Jobs for Delete Operations

Attribute	Description
IT Resource Name	Name of the IT resource for the target system installation from which you want to reconcile user records For DBUM Oracle Delete User Target Reconciliation: <code>Oracle DB</code> For DBUM Oracle Delete User Trusted Reconciliation, enter the name of the IT resource created for trusted source mode.
Object Type	Type of object you want to reconcile Default value: <code>User</code>

Table 4-15 (Cont.) Attributes of the Scheduled Jobs for Delete Operations

Attribute	Description
Resource Object Name	Name of the resource object that is used for reconciliation For DBUM Oracle Delete User Target Reconciliation: Oracle DB User For DBUM Oracle Delete User Trusted Reconciliation: Oracle DB Trusted

4.4.3 Configuring Scheduled Jobs for Oracle Database

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

See [Scheduled Jobs for Lookup Field Synchronization for Oracle Database](#) and [Attributes of the Scheduled Jobs for Oracle Database](#) for the scheduled jobs that are part of the connector and for information about their attributes.

To configure a scheduled job:

1. If you are using Oracle Identity Manager release 11.1.1.x:
 - a. Log in to the Administrative and User Console.
 - b. On the Welcome to Oracle Identity Manager Self Service page, click **Advanced** in the upper-right corner of the page.
 - c. On the Welcome to Oracle Identity Manager Advanced Administration page, in the System Management region, click **Search Scheduled Jobs**.
2. If you are using Oracle Identity Manager release 11.1.2.x or later:
 - a. Log in to Oracle Identity System Administration.
 - b. In the left pane, under System Management, click **Scheduler**.
3. Search for and open the scheduled job as follows:
 - a. On the left pane, in the Search field, enter the name of the scheduled job as the search criterion. Alternatively, you can click **Advanced Search** and specify the search criterion.
 - b. In the search results table on the left pane, click the scheduled job in the Job Name column.
4. On the Job Details tab, you can modify the following parameters:

Retries: Enter an integer value in this field. This number represents the number of times the scheduler tries to start the job before assigning the Stopped status to the job.

Schedule Type: Depending on the frequency at which you want the job to run, select the appropriate schedule type.

 **Note:**

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

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

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

 **Note:**

- Attribute values are predefined in the connector XML file that you import. Specify values only for those attributes that you want to change.
- Attributes of the scheduled job are discussed in [Attributes of the Scheduled Jobs for Oracle Database](#).

6. After specifying the attributes, click **Apply** to save the changes.

4.5 Reconciliation from Oracle Database

Postinstallation steps are divided across the following sections:

As mentioned earlier in this guide, reconciliation involves duplicating in Oracle Identity Manager the creation of and modifications to user accounts on the target system. This section discusses the following topics related to configuring reconciliation:

- [Guidelines on Configuring Reconciliation for Oracle Database](#)
- [About Reconciliation Process for Oracle Database](#)
- [About Reconciliation Queries for Oracle Database](#)
- [Target System Columns Used in Reconciliation from Oracle Database](#)
- [Configuring the Target System As a Trusted Source](#)
- [Reconciliation Rules for Oracle Database](#)
- [Viewing Reconciliation Rules for Oracle Database](#)
- [Reconciliation Action Rules for Oracle Database](#)
- [Viewing Reconciliation Action Rules for Oracle Database](#)
- [Performing Full Reconciliation from Oracle Database](#)
- [Performing Limited Reconciliation from Oracle Database](#)
- [Performing Limited Reconciliation from Oracle Database](#)
- [Performing Batched Reconciliation from Oracle Database](#)
- [Performing Incremental Reconciliation from Oracle Database](#)

4.5.1 Guidelines on Configuring Reconciliation for Oracle Database

These are the guidelines that you must apply while configuring reconciliation.

- Before you perform a target resource reconciliation run, you must synchronize the lookup definitions with the lookup fields of the target system. In other words, the scheduled job for lookup field synchronization must be run before user reconciliation runs.

- After you configure batched reconciliation, if reconciliation fails during a batched reconciliation run, then rerun the scheduled job without changing the values of the task attributes.

4.5.2 About Reconciliation Process for Oracle Database

This connector can be configured to perform either trusted source reconciliation or target resource reconciliation.

See Also:

Reconciliation Based on the Object Being Reconciled in *Oracle Fusion Middleware Administering Oracle Identity Manager* for conceptual information about target resource reconciliation and trusted source reconciliation.

When you configure the target system as a target resource, the connector enables you to create and manage database accounts for OIM Users through provisioning. In addition, data related to newly created and modified target system accounts can be reconciled and linked with existing OIM Users and provisioned resources.

When you configure the target system as a trusted source, the connector fetches into Oracle Identity Manager, data about newly created target system accounts. This data is used to create OIM Users.

Note:

During incremental reconciliation, only data about newly created accounts is available. Due to a limitation of the target system, the modified data is not part of the incremental updates.

The following is an overview of the steps involved in reconciliation:

1. A SQL query or stored procedure is used to fetch target system records during reconciliation.
2. The scheduled job communicates to connector bundle and runs search operations over it, maps the task attributes to parameters of the reconciliation query or stored procedure, and then runs the query or stored procedure on the target system.
3. Target system records that meet the query or stored procedure criteria are fetched into Oracle Identity Manager.
4. If you have configured your target system as a trusted source, then each user record fetched from the target system is compared with existing OIM Users. The reconciliation rule is applied during the comparison process.

The next step of the process depends on the outcome of the matching operation:

- If a match is found between the target system record and the OIM User, then the OIM User attributes are updated with changes made to the target system record.

- If no match is found, then the target system record is used to create an OIM User.
5. If you have configured your target system as a target resource, then each user record fetched from the target system is compared with existing target system resources assigned to OIM Users. The reconciliation rule is applied during the comparison process.

The next step of the process depends on the outcome of the matching operation:

- If a match is found between the target system record and a resource provisioned to an OIM User, then the database user resource is updated with changes made to the target system record.
- If no match is found, then the target system user record is compared with existing OIM Users. The next step depends on the outcome of the matching operation:

If a match is found, then the target system record is used to provision a resource for the OIM User.

If no match is found, then the status of the reconciliation event is set to No Match Found.

**Note:**

[Reconciliation Rules for Oracle Database](#) for information about the reconciliation rule

4.5.3 About Reconciliation Queries for Oracle Database

As mentioned earlier in this chapter, a SQL query or a stored procedure is used to fetch target system records during reconciliation. All predefined SQL queries and stored procedures are stored in a JAR file in the bundle directory of the connector installation media.

For example, to locate the reconciliation query file, you can extract the `bundle/org.identityconnectors.dbum-1.0.1116.jar` file and open `scripts/oracle/Search.queries`.

**Note:**

Depending on your requirements, you can modify existing queries or add your own query in the query file. Alternatively, you can create and use your own query file. [About the Queries for Oracle Database](#) provides more information.

Some of the predefined queries for Oracle Database are used in conjunction with the Incremental Recon Attribute scheduled job attribute. This attribute stores the time stamp at which the last reconciliation run started. When the next reconciliation run begins, only target system records for which the `lastModified` column value is greater than the value of the Incremental Recon Attribute are fetched into Oracle Identity Manager. In other words, only records that were added or modified after the last reconciliation run started are considered for the current reconciliation run.

 **Note:**

Update operations for Oracle Database users are processed based on the create time-stamp, which is assigned to a user when the user is created. During incremental reconciliation, only the users created after this time-stamp are fetched. However, the users updated after the time-stamp are not fetched.

The following are the predefined queries for Oracle Database:

- **SEARCH_USER**
This query is used to fetch all user records from the `DBA_USERS` table.
- **BATCHED_SEARCH_USER**
This query is used to fetch from the `DBA_USERS` table user records that are present within the specified range. It is used to perform batched reconciliation on a target system that is configured as a target resource.
- **SEARCH_USER_ROLE**
This query is used to fetch all user roles from the `DBA_ROLE_PRIVS` table.
- **SEARCH_USER_PRIVILEGE**
This query is used to fetch all user privileges from the `DBA_SYS_PRIVS` table.

4.5.4 Target System Columns Used in Reconciliation from Oracle Database

As mentioned earlier in this guide, this connector can be configured to perform either target resource reconciliation or trusted source reconciliation. This section discusses the following topics:

- The `Lookup.DBUM.Oracle.UM.ReconAttrMap` lookup definition holds attribute mappings for user reconciliation. This lookup definition contains mapping of Oracle Identity Manager attributes and connector attributes.
See [Lookup.DBUM.Oracle.UM.ReconAttrMap](#) for more information.
- The `Lookup.DBUM.Oracle.UM.ReconAttrMap.Trusted` lookup definition holds attribute mappings for reconciliation in trusted mode. This lookup definition maps reconciliation profile attributes and connector attributes used in the reconciliation query. In addition, the connector attributes are associated to columns within the bundle.
See [Lookup.DBUM.Oracle.UM.ReconAttrMap.Trusted](#) for more information about this lookup definition.

4.5.5 Configuring the Target System As a Trusted Source



Note:

Skip this section if you do not want to designate the target system as a trusted source for reconciliation.

To configure trusted source reconciliation:

1. If you are using Oracle Identity Manager release 11.1.1.x:
 - a. Log in to the Administrative and User Console.
 - b. On the Welcome page, click **Advanced** in the upper-right corner of the page.
 - c. On the Welcome to Oracle Identity Manager Advanced Administration page, in the Configuration region, click **Create IT Resource**.
2. If you are using Oracle Identity Manager release 11.1.2.x or later:
 - a. Log in to Oracle Identity System Administration.
 - b. In the left pane, under Configuration, click **IT Resource**.
 - c. In the Manage IT Resource page, click **Create IT Resource**.
3. On the Step 1: Provide IT Resource Information page, enter the following information:
 - **IT Resource Name:** Enter a name for the IT resource. For example, `Oracle DB Trusted`.
 - **IT Resource Type:** Select the **Oracle DB** IT resource type for the IT resource.
4. Click **Continue**.
5. On the Step 2: Specify IT Resource Parameter Values page, specify values for the parameters of the IT resource.

Configuration Lookup: Name of the lookup definition in which you store the connector configuration information for the target system.

Sample Value: `Lookup.DBUM.Oracle.Configuration.Trusted`

Provide values for the other IT resource parameters.

6. Click **Continue**.

In the following steps, provide permissions on the IT resource that you are creating as per your requirements.

You can use this IT resource for trusted source reconciliation operations.

4.5.6 Reconciliation Rules for Oracle Database

See Also:

Reconciliation Metadata in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for generic information about reconciliation rules and reconciliation action rules

This section describes the reconciliation rules used by the reconciliation engine for this connector.

The following are the reconciliation rules for target resource reconciliation:

- **Rule name:** DBUM Oracle Target Recon
- **Rule element:** User Login Equals User Name

The following are the reconciliation rules for trusted source reconciliation:

- **Rule name:** Oracle DB Trusted
- **Rule element:** User Login Equal User ID

In these rule elements:

- User Login is the field on the OIM User form.
- User Name and User ID are the target system fields.

4.5.7 Viewing Reconciliation Rules for Oracle Database

After you deploy the connector, you can view the reconciliation rule for reconciliation by performing the following steps:

Note:

Perform the following procedure only after the connector is deployed.

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Development Tools**.
3. Double-click **Reconciliation Rules**.
4. Search for the rule name.

4.5.8 Reconciliation Action Rules for Oracle Database

Reconciliation action rules define that actions the connector must perform based on the reconciliation rules defined for Users.

[Table 4-16](#) lists the action rules for target resource reconciliation.

Table 4-16 Action Rules for Target Resource Reconciliation

Rule Condition	Action
No Matches Found	Assign to Administrator With Least Load
One Entity Match Found	Establish Link
One Process Match Found	Establish Link

[Table 4-17](#) lists the action rules for trusted source reconciliation.

Table 4-17 Action Rules for Trusted Source Reconciliation

Rule Condition	Action
No Matches Found	Create User
One Entity Match Found	Establish Link

4.5.9 Viewing Reconciliation Action Rules for Oracle Database

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

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Resource Management**.
3. Double-click **Resource Objects**.
4. Search for and open the resource object. The following are the names of the resource objects for each target system database:
 - Resource object for Oracle Database:
Oracle DB User
 - Resource object for Oracle Database as trusted source:
Oracle DB Trusted
5. Click the **Object Reconciliation** tab, and then click the **Reconciliation Action Rules** tab. The Reconciliation Action Rules tab displays the action rules defined for this connector.

4.5.10 Performing Full Reconciliation from Oracle Database

Full reconciliation involves reconciling all existing user records from the target system into Oracle Identity Manager. After you deploy the connector, you must first perform full reconciliation.

To perform a full reconciliation run, remove (delete) any value currently assigned to the Filter attribute and run one of the following scheduled jobs:

- For Oracle Database as a target resource: DBUM Oracle User Target Reconciliation
- For Oracle Database as a trusted source: DBUM Oracle User Trusted Reconciliation

See [Attributes of the Scheduled Jobs for Oracle Database](#) for information about this scheduled job.

4.5.11 Performing Limited Reconciliation from Oracle Database

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

You can perform limited reconciliation by creating filters for the reconciliation module. This connector provides a Filter attribute (a scheduled task attribute) that allows you to use any of the DBUM resource attributes to filter the target system records. You can apply filters to the parent parameters in the reconciliation query file stored in a JAR file in the bundle directory of the connector installation media. For example, to locate the reconciliation query file, you can extract the `bundle/org.identityconnectors.dbum-1.0.1116.jar` file and open `scripts/oracle/Search.queries`.

The following table provides a list of parent parameters that can be used with the Filter attribute of the scheduled jobs:

Parameter	Description
__UID__	Unique identity representing the user This parameter is mapped to USERNAME or __NAME__ connector attribute.
authType	Authentication type of the user account The value of this parameter can be one of the following: PASSWORD, GLOBAL, or EXTERNAL
tablespace	Default tablespace for user operations
defaultQuota	Quota for user operations on default tablespace If no value is specified, the quota is set to unlimited.
globalDN	Unique name that identifies a user across an enterprise, if the authentication type is GLOBAL
__ENABLE__	Status of the user account The user is disabled if the value is one of following: LOCKED, EXPIRED, or LOCKED & EXPIRED The list of values for the disabled status is provided in the Lookup.DBUM.Oracle.Configuration lookup definition.
tempTableSpace	Temporary tablespace for user operations Quota is always unlimited on temporary tablespace.
profile	Profile of the user account
lastModified	Last modified time-stamp This parameter is used for incremental reconciliation operations.

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

While deploying the connector, follow the instructions in [Configuring Scheduled Jobs for Oracle Database](#) to specify attribute values.

4.5.12 Performing Batched Reconciliation from Oracle Database

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

You can configure batched reconciliation to avoid these problems.

To configure batched reconciliation, you must specify value for the Batch Size reconciliation scheduled job attribute. Use this attribute to specify the number of records that must be included in each batch. By default, this value is empty.

If you specify a value other than `All`, then some of the newly added or modified user records may not get reconciled during the current reconciliation run. The following example illustrates this:

Suppose you specify the Batch Size value as `200` while configuring the scheduled jobs. Suppose that 314 user records were created or modified after the last reconciliation run. Of these 314 records, only 200 records would be reconciled during the current reconciliation run. The remaining 114 records would be reconciled during the next reconciliation run.

You specify values for the Batch Size attribute by following the instructions described in [Configuring Scheduled Jobs for Oracle Database](#).

4.5.13 Performing Incremental Reconciliation from Oracle Database

During an incremental reconciliation run, the scheduled job fetches only target system records that are added or modified after the time-stamp stored in the Latest Token attribute of the scheduled job. The connector requires a query to calculate the time-stamp value. This time-stamp value is used by the query that is used to perform reconciliation.

Note:

Update operations for Oracle Database users are processed based on the create time-stamp, which is assigned to a user when the user is created. During incremental reconciliation, only the users created after this time-stamp are fetched. However, the users updated after the time-stamp are not fetched.

4.6 Provisioning for Oracle Database

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

This section contains the following topics about provisioning:

- [Guidelines on Performing Provisioning Operations for Oracle Database](#)
- [Understanding the Provisioning Process for Oracle Database](#)
- [Configuring Direct Provisioning for Oracle Database](#)
- [Configuring Request-Based Provisioning for Oracle Database](#)

- [Switching Between Request-Based Provisioning and Direct Provisioning for Oracle Database](#)
- [Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x](#)

4.6.1 Guidelines on Performing Provisioning Operations for Oracle Database

The following are guidelines that you must apply while performing provisioning operations:

- Before you perform provisioning operations, lookup definitions must be synchronized with the lookup fields of the target system. In other words, run the scheduled jobs for lookup field synchronization before provisioning operations.
- Passwords for user accounts provisioned from Oracle Identity Manager must adhere to the password policy set in the target system.
- The character length of target system fields must be taken into account when specifying values for the corresponding Oracle Identity Manager fields.
- During an update password provisioning operation, ensure that you clear the existing text in the Password field, and then enter the new password.
- During a Create User provisioning operation, the following are some of the fields that are optional:

- Default Tablespace
- Default Tablespace Quota (in MB)

This field is dependent on Default Tablespace. To specify a quota, you must specify a value for Default Tablespace.

- Temporary Tablespace
- Profile Name

If you specify a value for any of these fields during a Create User provisioning operation, then you must not leave them empty during an Update User provisioning operation. Otherwise, the provisioning operation will fail. However, you can modify the existing values in these fields.

- For creating password-authenticated database users, you must specify values for the following fields:
 - **IT Resource:** Specify `Oracle DB` as the value of this lookup field.
 - **Username:** Enter the name of the database user.
 - **Password:** Enter the password for the database user.
 - **Authentication Type:** Specify `PASSWORD` as the value of this lookup field.
- For creating globally-authenticated database users, you must specify a value for the following mandatory fields:
 - **IT Resource:** Specify `Oracle DB` as the value of this lookup field.
 - **Username:** Enter the name of the database user.
 - **Authentication Type:** Specify `GLOBAL` as the value of this lookup field.
 - **Global DN:** Enter the distinguished name (DN) for your organization.

Sample value: `cn=ajones,cn=users,dc=oracle,dc=vm`

After you submit the data required, the connector runs the following query to create a globally-authenticated database user:

```
CREATE USER {__NAME__} IDENTIFIED GLOBALLY AS {globalDN}
```

- For creating externally-authenticated database users, you must specify a value for the following mandatory fields:
 - **IT Resource:** Specify `Oracle DB` as the value of this lookup field.
 - **Username:** Enter the name of the database user.
 - **Authentication Type:** Specify `EXTERNAL` as the value of this lookup field.

After you submit the data required, the adapter runs the following query to create an externally-authenticated database user:

```
CREATE USER {__NAME__} IDENTIFIED EXTERNALLY
```

- If you specify a value for the Default Tablespace Quota (in MB) field, then enter values in the following format:

```
TABLESPACE_QUOTA M
```

In this format, `TABLESPACE_QUOTA` is the tablespace quota allocated to the user and `M` indicates that megabytes is the unit of measurement of quota. The following is a sample value: `300 M`

If you want to allocate to a user unlimited quota on a tablespace, then specify the following as the value of the Default Tablespace Quota (in MB) field:

```
UNLIMITED
```

4.6.2 Understanding the Provisioning Process for Oracle Database

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

When you install the connector on Oracle Identity Manager, the direct provisioning feature is automatically enabled. This means that the process form is enabled when you install the connector.

The following are types of provisioning operations:

- Direct provisioning
- Request-based provisioning
- Provisioning triggered by policy changes

If you configure the connector for request-based provisioning, then the process form is suppressed and the object form is displayed. In other words, direct provisioning is disabled when you configure the connector for request-based provisioning. If you want to revert to direct provisioning, then see [Switching Between Request-Based Provisioning and Direct Provisioning for Oracle Database](#).

The following is an overview of the Create User provisioning process in Oracle Database that is started through direct provisioning:

1. On the Create User page of the Administrative and User Console, the administrator enters the data required for an OIM User account creation.

Suppose the administrator enters the following values for the fields on the Create User page:

- First Name: John
- Last Name: Doe
- User ID: jdoe

An OIM User account is created for John Doe.

2. The administrator selects the resource to be provisioned to the OIM User account that has been created. In this example, the administrator selects the Oracle DB User resource.
3. The administrator enters the data required for provisioning the Oracle DB User resource. Suppose the administrator wants to create a local user that requires a password to log in to the database. Therefore, the administrator enters the following values on the resource provisioning process form:

- IT Resource: Oracle DB
- Username: JDoe
- Authentication Type: PASSWORD
- Password: my_pa55word
- Default Tablespace: example
- Profile Name: dba_user

In addition, the administrator also enters the following values on the process form for granting roles:

- Role: 3~JAVA_ADMIN
- Role Admin Option: WITH ADMIN OPTION

4. From the information available in the IT resource for the target system, the configuration (Lookup.DBUM.Oracle.Configuration) lookup definition is identified. This lookup definition stores configuration information that is used during connector operations.
5. The connector bundle contains the script (Provisioning.queries) required for provisioning operations.
6. The identifiers in the SQL statement are replaced with the input parameters fetched from the query. Then, the SQL statement with actual values is formed.

Suppose while performing Step 1, the administrator enters `jdoe` as the value of the User ID field. While performing Step 3 of this procedure, the Username field is prepopulated with the value that the administrator had entered in the User ID field. Now, suppose while performing Step 3 of this procedure, the administrator enters `example` and `dba_users` as the values of the Default Table Space and Profile Name process form fields, respectively. The SQL statement with the actual values is as follows:

```
CREATE USER jdoe IDENTIFIED BY dba_users ACCOUNT UNLOCK DEFAULT
TABLE SPACE example PROFILE db_user
```

7. The connector runs the SQL statement on Oracle Database and creates the jdoe account on the target system. The next step of the process depends on whether the administrator had entered data for granting roles or privileges to the target system account.

If the administrator did not enter any values for granting roles, then the provisioning process ends here. Otherwise, the process continues to the next step.
8. While performing Step 3, the administrator had entered the required data for granting roles to the jdoe account. Therefore, the corresponding query as mentioned in Step 6 is read.
9. The complete SQL statement that must be run to perform the Add role provisioning operation is formed. Depending on whether the administrator had granted a role with the admin option, the SQL statement is one of the following:
 - If the administrator specified a value for granting the role with the admin option, then the following SQL statement is formed:

```
GRANT JAVA_ADMIN TO jdoe WITH ADMIN OPTION
```
 - If the administrator did not specify a value for granting role with the admin option, then the following SQL statement is formed:

```
GRANT JAVA_ADMIN TO jdoe
```
10. The input parameters required to run the SQL statement are fetched from the parameter configuration done using the queries in the query files.
11. The identifiers in the SQL statement (formed in Step 11) are replaced with the input parameters fetched from the query. Then, the SQL statement with actual values is formed.
12. The query runs the SQL statement on the target system (Oracle database) and grants the role JAVA_ADMIN to the jdoe target system account.

4.6.3 Configuring Direct Provisioning for Oracle Database

In direct provisioning, the Oracle Identity Manager administrator uses the Administrative and User Console to create a target system account for a user.

To provision a resource by using the direct provisioning approach:

1. Log in to the Administrative and User Console.
2. To first create an OIM User before provisioning a database account to the user:
 - a. On the Welcome to Identity Administration page, in the Users region, click **Create User**.
 - b. On the Create User page, enter values for the OIM User fields, and then click the save icon.
3. To search for an existing OIM User to be provisioned:
 - a. On the Welcome to Identity Administration page, search for the user by selecting **Users** from the Search list on the left pane.

Alternatively, in the Users region, click **Advanced Search - User**, provide a search criterion, and then click **Search**.
 - b. From the list of users displayed in the search results, select the OIM User.

The user details page is displayed.

4. From the Action menu, select **Add Resource**. Alternatively, you can click the add resource icon with the plus (+) sign. The Provision Resource to User page is displayed in a new window.
5. On the Step 1: Select a Resource page, select the **Oracle DB User** resource from the list, and then click **Continue**.
6. On the Step 2: Verify Resource Selection page, click **Continue**.
7. On the Step 5: Provide Process Data page, enter the details of the account that you want to create on the target system and then click **Continue**.
8. If you want to provide child data, then on the Step 5: Provide Process Data page for child data, search for and select the child data for the user on the target system and then click **Continue**. Repeat the same step if you have more than one child data and you want to provision them.
9. On the Step 6: Verify Process Data page, verify the data that you have provided and then click **Continue**.
10. The "Provisioning has been initiated" message is displayed. Perform the following steps:
 - a. Close the window displaying the "Provisioning has been initiated" message.
 - b. On the Resources tab, click **Refresh** to view the newly provisioned resource.If the resource status is Provisioned, then provisioning was successful. If the status is Provisioning, then there may be an error. To verify if there was an error, you can check the resource history.

4.6.4 Configuring Request-Based Provisioning for Oracle Database

The following sections discuss the steps to be performed to enable request-based provisioning:

- [About Request-Based Provisioning for Oracle Database](#)
- [Enabling Request-Based Provisioning](#)

4.6.4.1 About Request-Based Provisioning for Oracle Database

In request-based provisioning, an end user creates a request for a resource by using the Administrative and User Console. Administrators or other users can also create requests for a particular user. Requests for a particular resource on the resource can be viewed and approved by approvers designated in Oracle Identity Manager.

The following are features of request-based provisioning:

- A user can be provisioned only one resource (account) on the target system.

 **Note:**

Direct provisioning allows the provisioning of multiple database accounts on the target system.

- Direct provisioning cannot be used if you enable request-based provisioning.

4.6.4.2 Enabling Request-Based Provisioning

The following sections provide information about the procedures you must perform to enable request-based provisioning:



Note:

The procedure described in this section is applicable only if you are using Oracle Identity Manager release 11.1.1.x.

- [Approver's Role in Request-Based Provisioning](#)
- [Importing Request Datasets Using Deployment Manager](#)
- [End User's Role in Request-Based Provisioning](#)
- [Enabling the Auto Save Form Feature](#)
- [Running the PurgeCache Utility](#)

4.6.4.2.1 Approver's Role in Request-Based Provisioning

The following are steps performed by the approver in a request-based provisioning operation:

1. Log in to the Administrative and User Console.
2. On the Welcome page, click **Self-Service** in the upper-right corner of the page.
3. On the Welcome to Identity Manager Self Service page, click the **Tasks** tab.
4. On the **Approvals** tab, in the first section, you can specify a search criterion for request task that is assigned to you.
5. From the search results table, select the row containing the request you want to approve, and then click **Approve Task**.

A message confirming that the task was approved is displayed.

4.6.4.2.2 Importing Request Datasets Using Deployment Manager

A request dataset is an XML file that specifies the information to be submitted by the requester during a provisioning operation. These request datasets specify information about the default set of attributes for which the requester must submit information during a request-based provisioning operation.

To import a request dataset XML file by using the Deployment Manager:

1. Log in to the Oracle Identity Manager Administrative and User Console.
2. Click the **Deployment Management** link on the left navigation bar.
3. Click the **Import** link under Deployment Management.

A dialog box for opening files is displayed.

4. Locate and open the request dataset XML file, `DBUserManagement-Oracle-Datasets.xml`, which is in the `xml` directory of the installation media.

Details of this XML file are shown on the **File Preview** page.

5. Click **Add File**.
The Substitutions page is displayed.
6. Click **Next**.
The Confirmation page is displayed.
7. Click **Import**.
8. Close the Deployment Manager dialog box.
The request dataset is imported into Oracle Identity Manager.

4.6.4.2.3 End User's Role in Request-Based Provisioning

The following steps are performed by the end user in a request-based provisioning operation:

1. Log in to the Administrative and User Console.
2. On the Welcome page, click **Advanced** in the upper-right corner of the page.
3. On the Welcome to Identity Administration page, click the **Administration** tab, and then click the **Requests** tab.
4. From the Actions menu on the left pane, select **Create Request**.
The Select Request Template page is displayed.
5. From the Request Template list, select **Provision Resource** and click **Next**.
6. On the Select Users page, specify a search criterion in the fields to search for the user that you want to provision the resource, and then click **Search**. A list of users that match the search criterion you specify is displayed in the Available Users list.
7. From the **Available Users** list, select the user to whom you want to provision the account.

If you want to create a provisioning request for more than one user, then from the **Available Users** list, select users to whom you want to provision the account.
8. Click **Move** or **Move All** to include your selection in the Selected Users list, and then click **Next**.
9. On the Select Resources page, click the arrow button next to the Resource Name field to display the list of all available resources.
10. From the Available Resources list, select **Oracle DB User**, move it to the Selected Resources list, and then click **Next**.
11. On the Resource Details page, enter details of the account that must be created on the target system, and then click **Next**.
12. On the Justification page, you can specify values for the following fields, and then click **Finish**.
 - Effective Date
 - Justification
A message confirming that your request has been sent successfully is displayed along with the Request ID.
13. If you click the request ID, then the Request Details page is displayed.

14. To view details of the approval, on the Request Details page, click the **Request History** tab.

4.6.4.2.4 Enabling the Auto Save Form Feature

To enable the Auto Save Form feature:

1. Log in to the Design Console.
2. Expand **Process Management**, and then double-click **Process Definition**.
3. Search for and open the **Oracle DB** process definition.
4. Select the **Auto Save Form** check box.
5. Click the save icon.

4.6.4.2.5 Running the PurgeCache Utility

Run the PurgeCache utility to clear content belonging to the Metadata category from the server cache.

See [Clearing Content Related to the Connector Resource Bundles from the Server Cache](#) for instructions.

The procedure to enable enabling request-based provisioning ends with this step.

4.6.5 Switching Between Request-Based Provisioning and Direct Provisioning for Oracle Database

If you have configured the connector for request-based provisioning, you can always switch to direct provisioning. Similarly, you can always switch back to request-based provisioning any time. This section discusses the following topics:

- [Switching From Request-Based Provisioning to Direct Provisioning](#)
- [Switching From Direct Provisioning to Request-Based Provisioning](#)

4.6.5.1 Switching From Request-Based Provisioning to Direct Provisioning

**Note:**

It is assumed that you have performed the procedure described in [Configuring Request-Based Provisioning for Oracle Database](#).

To switch from request-based provisioning to direct provisioning:

1. Log in to the Design Console.
2. Disable the Auto Save Form feature as follows:
 - a. Expand **Process Management**, and then double-click **Process Definition**.
 - b. Search for and open the **Oracle DB** process definition.
 - c. Deselect the Auto Save Form check box.

- d. Click the save icon.
3. If the Self Request Allowed feature is enabled, then:
 - a. Expand **Resource Management**, and then double-click **Resource Objects**.
 - b. Search for and open the **Oracle DB User** resource object.
 - c. Deselect the Self Request Allowed check box.
 - d. Click the save icon.

4.6.5.2 Switching From Direct Provisioning to Request-Based Provisioning

To switch from direct provisioning back to request-based provisioning:

1. Log in to the Design Console.
2. Enable the Auto Save Form feature as follows:
 - a. Expand **Process Management**, and then double-click **Process Definition**.
 - b. Search for and open the **Oracle DB** process definition.
 - c. Select the **Auto Save Form** check box.
 - d. Click the save icon.
3. If you want to enable end users to raise requests for themselves, then:
 - a. Expand **Resource Management**, and then double-click **Resource Objects**.
 - b. Search for and open the **Oracle DB User** resource object.
 - c. Select the Self Request Allowed check box.
 - d. Click the save icon.

4.6.6 Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x

To perform provisioning operations in Oracle Identity Manager release 11.1.2.x:

1. Log in to Identity System Administration.
2. If you want to first create an OIM User and then provision a target system account, then:

 **Note:**

See *Creating Users in Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Manager* for more information about creating a user.

- a. In the left pane, under Administration, click **Users**.
The Search Users page is displayed.
- b. From the Actions menu, select **Create**. Alternatively, you can click **Create** on the toolbar.

- c. On the Create User page, enter values for the OIM User fields, and then click **Submit**. A message is displayed stating that the user is created successfully.
3. If you want to provision a target system account to an existing OIM User, then:

 **Note:**

See Searching Users in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Manager* for more information about searching a user.

- a. In the left pane, under Administration, click **Users**.
The Search Users page is displayed.
 - b. Specify a search criteria to search for the OIM User, and then click **Search**.
 - c. From the list of users displayed in the search results, select the OIM User. The user details page is displayed on the right pane.
4. On the Account tab, click **Request Accounts**.
5. In the Catalog page, search for and add to cart the application instance (in other words, the account to be provisioned), and then click **Checkout**.
6. Specify value for fields in the application form and then click **Ready to Submit**.
7. Click **Submit**.
8. If you want to provision entitlements, then:
 - a. On the Entitlements tab, click **Request Entitlements**.
 - b. In the Catalog page, search for and add to cart the entitlement, and then click **Checkout**.
 - c. Click **Submit**.

4.7 Extending the Connector for Oracle Database

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

The section contains the following topics:

 **Note:**

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

- [Modifying the Predefined Queries or Creating New Queries](#)
- [Configuring Queries to Add Support for Custom Parameters and Lookup Fields for Oracle Database](#)

- [About Configuring the Connector for Multiple Installations of Oracle Database](#)
- [About Configuring the Connector for Multiple Trusted Source Reconciliation from Oracle Database](#)
- [Configuring Validation of Data During Reconciliation and Provisioning for Oracle Database](#)
- [Configuring Transformation of Data During User Reconciliation for Oracle Database](#)
- [Configuring Resource Exclusion Lists for Oracle Database](#)
- [Configuring Action Scripts for Oracle Database](#)

4.7.1 Modifying the Predefined Queries or Creating New Queries

The following sections discuss the syntax and guidelines that you must apply while modifying the predefined queries or creating new queries:

- [About the Queries for Oracle Database](#)
- [Syntax of Provisioning Queries for Oracle Database](#)
- [Syntax of Reconciliation Queries for Oracle Database](#)
- [Syntax of List of Values Queries for Oracle Database](#)
- [Guidelines for Configuring Search Queries Used in Reconciliation from Oracle DatabaseS](#)

4.7.1.1 About the Queries for Oracle Database

Predefined queries are provided to reconcile target system user records, synchronize lookup field values with Oracle Identity Manager, and for provisioning operations. You can modify the predefined queries or add your own queries.

The query files are included in a JAR file in the bundle directory of the connector installation media. For example, `bundle/org.identityconnectors.dbum-1.0.1116.jar`.

The connector includes the following types of queries:

- **Provisioning Queries**
They are used for create, update, and delete operations. The query file is `scripts/oracle/Provisioning.queries`.
- **List of Values Search Queries**
They are used for reconciliation of lookup definitions. A list of value query operates on a set of values for fields such as profiles, privileges, roles, and tablespaces. The query file is `scripts/oracle/LoVSearch.queries`.
- **Account Search Queries**
They are used for full, incremental, and delete reconciliation operations. An account search query operates on account and group searches with various conditions. The query file is `scripts/oracle/Search.queries`.

 **Note:**

The stored procedure OUT parameters cannot be configured for write-back on the process form. The returned values cannot be used for any connector operations.

4.7.1.2 Syntax of Provisioning Queries for Oracle Database

The following is the syntax of the queries used for provisioning operations:

```

QUERYID {
Query="QUERY"
QueryType="QUERYTYPE"
Parameters=["PARAM1":"PARAMDEFN1", "PARAM2":"PARAMDEFN2" ...]
ExtensionJoin="EXTENSIONJOIN"
ExtensionSeparator="EXTENSIONSEPARATOR"
QueryExtensions=["EXTENSION1", "EXTENSION2" ...]
}

```

For example:

```

CREATE_EXTERNAL_USER {
  Query="CREATE USER {__NAME__} IDENTIFIED EXTERNALLY"
  QueryType="SQL"
  Parameters=["__NAME__": "Type:String, TAGS:DOUBLEQUOTES"]
  ExtensionJoin=", "
  ExtensionSeparator=", "
  QueryExtensions=["TEMP_TABLESPACE_QUERY", "TABLESPACE_QUERY", "PROFILE_QUERY"]
}

```

In this syntax:

- *QUERYID* refers to the unique name of the query.

For example: `CREATE_EXTERNAL_USER`

For `CREATE` provisioning queries, the format of *QUERYID* is `CREATE_AUTHENTICATIONTYPE_ACCOUNTTYPE`. The default account type is `USER`. For other provisioning queries, the format is the `OPERATIONTYPE_ATTRIBUTE`, such as `UPDATE_GLOBALDN`.

- *QUERY* refers to the main query.

For example: `Query="CREATE USER {__NAME__} IDENTIFIED EXTERNALLY"`

- *QueryType* refers to the type of the main query, either an SQL query or a stored procedure. The value of *QUERYTYPE* can be `SQL` or `StoredProc`.

For example: `QueryType="SQL"`

- *Parameters* refers to the list of comma separated parameters and parameter definitions used with the main query, represented by `"PARAM1":"PARAMDEFN1"`, `"PARAM2":"PARAMDEFN2"`, and so on.

For example: Parameters=["__NAME__": "Type:String, TAGS:DOUBLEQUOTES"]

A parameter can have the following attributes:

- Type is the type of the parameter.
- Direction is the flow of data from the query to or from the parameter. It can have a value of IN, OUT, or INOUT.
- TAGS is the enclosure characters that are applied to each parameter before the query is processed. It can have a value of DOUBLEQUOTES, QUOTES, UPPERCASE, or LOWERCASE.

If you want to use multiple tags, you must encapsulate the tags in escaped quotes and separate them by commas. However, you must not use DOUBLEQUOTES with QUOTES or UPPERCASE with LOWERCASE in the same query.

For example: "Type:String, TAGS:\ "DOUBLEQUOTES, UPPERCASE\ "

- ExtensionJoin (optional) refers to the operator, represented by *EXTENSIONJOIN*, used to join the main query with query extensions.

For example: ExtensionJoin=","

- ExtensionSeparator (optional) refers to the delimiter between query extensions, represented by *EXTENSIONSEPARATOR*.

For example: ExtensionSeparator=","

- QueryExtensions (optional) refers to the extensions that must be appended to the main query, represented by *EXTENSION1*, *EXTENSION2*, and so on.

For example:

```
QueryExtensions=["TEMP_TABLESPACE_QUERY", "TABLESPACE_QUERY", "PROFILE_QUERY"]
```

During a provisioning operation, the connector combines all these components to the following query:

```
QUERY PARAM1, PARAM2... [EXTENSIONJOIN [EXTENSION1  

EXTENSIONSEPARATOR EXTENSION2 EXTENSIONSEPARATOR...]]
```

For example:

```
CREATE USER {__NAME__} IDENTIFIED EXTERNALLY, TEMP_TABLESPACE_QUERY,  

TABLESPACE_QUERY, PROFILE_QUERY
```

Table 4-18 lists the script selection logic of the provisioning queries:

Table 4-18 Script Section Logic for Oracle Provisioning Queries

Operation	Selection Logic	Query IDs
CREATE	CREATE_AUTHTYPE_OBJECTTYPE	CREATE_PASSWORD_USER CREATE_GLOBAL_USER CREATE_EXTERNAL_USER
DELETE	DELETE_OBJECTTYPE	DELETE_USER
ENABLE	ENABLE_OBJECTTYPE	ENABLE_USER
DISABLE	DISABLE_OBJECTTYPE	DISABLE_USER
RESET PASSWORD	SET_PASSWORD	SET_PASSWORD

Table 4-18 (Cont.) Script Section Logic for Oracle Provisioning Queries

Operation	Selection Logic	Query IDs
UPDATE	UPDATE_ATTRIBUTE	UPDATE_TABLESPACE UPDATE_DEFAULTQUOTA UPDATE_GLOBALDN UPDATE_PROFILE UPDATE_TEMP_TABLESPACE
ADD CHILD VALUES	UPDATE_ADD_ATTRIBUTE	UPDATE_ADD_ROLES UPDATE_ADD_PRIVILEGES
REMOVE CHILD VALUES	UPDATE_REVOKE_ATTRIBUTE	UPDATE_REVOKE_ROLES UPDATE_REVOKE_PRIVILEGES

4.7.1.3 Syntax of Reconciliation Queries for Oracle Database

The following is the syntax of the search queries used during reconciliation operations:

```

QUERYID {
  Query="QUERY"
  QueryType="QUERYTYPE"
  Parameters=["PARAM1":"PARAMDEFN1", "PARAM2":"PARAMDEFN2"...]
  ExtensionJoin="EXTENSIONJOIN"
  ExtensionSeparator="EXTENSIONSEPARATOR"
  QueryExtensions=["EXTENSION1","EXTENSION2"...]
}
    
```

For example:

```

SEARCH_USER {
  Query="SELECT {__UID__}, {authType}, {externalname}, {tablespace}, {status},
  {tempTableSpace}, {profile}," +
  " {defaultQuota}, {tmpQuota}, {lastModified} FROM DBA_USERS dba {filter}"
  QueryType="SQL"
  Parameters=["__UID__":"Type:String,Direction:OUT,ColName:USERNAME",

  "authType":"Type:String,Direction:OUT,ColName:PASSWORD,ColQuery:\\"DECODE (PASSWORD,
  'EXTERNAL', 'EXTERNAL', 'GLOBAL', 'GLOBAL', 'PASSWORD')\\"",
  "tablespace":"Type:String,Direction:OUT,ColName:DEFAULT_TABLESPACE",

  "tmpQuota":"Type:String,Direction:OUT,ColName:TEMPORARY_TABLESPACE_QUOTA,ColQuery:
  (SELECT MAX_BYTES FROM DBA_TS_QUOTAS WHERE dba.USERNAME = USERNAME AND TABLESPACE_NAME
  = dba.TEMPORARY_TABLESPACE)",

  "defaultQuota":"Type:String,Direction:OUT,ColName:DEFAULT_TABLESPACE_QUOTA,ColQuery:
  (SELECT MAX_BYTES FROM DBA_TS_QUOTAS WHERE dba.USERNAME = USERNAME AND TABLESPACE_NAME
  = dba.DEFAULT_TABLESPACE)",
  "externalname":"Type:String,Direction:OUT,ColName:EXTERNAL_NAME",
  "status":"Type:String,Direction:OUT,ColName:ACCOUNT_STATUS",
  "tempTableSpace":"Type:String,Direction:OUT,ColName:TEMPORARY_TABLESPACE",
    
```

```

    "profile":"Type:String,Direction:OUT,ColName:PROFILE",
    "lastModified":"Type:long,Direction:OUT,ColName:TIMESTAMP,
ColQuery:\\"((CREATED - TO_DATE('01011970','ddmmyyy')) *24*60*60*1000)\\""]
    QueryExtensions=["SEARCH_USER_ROLE", "SEARCH_USER_PRIVILEGE"]
  }

```

In this syntax:

- **QUERYID** refers to the unique name of the query.

For example: SEARCH_USER

QUERYID can be one of the following values:

- SEARCH_USER
- BATCHED_SEARCH_USER
- SEARCH_USER_ROLE
- SEARCH_USER_PRIVILEGE

- **QUERY** refers to the main query.

For example: Query="SELECT {__UID__}, {authType}, {externalname}, {tablespace}, {status}, {tempTableSpace}, {profile}," +
 " {defaultQuota}, {tmpQuota}, {lastModified} FROM DBA_USERS dba {filter}"

- **QueryType** refers to the type of the main query, either an SQL query, a stored procedure, or a query extension. The value of **QUERYTYPE** can be SQL, StoredProc, or QUERYEXTENSION.

For example: QueryType="SQL"

- **Parameters** refers to the list of comma separated parameters and parameter definitions used with the main query, represented by "PARAM1":"PARAMDEFN1", "PARAM2":"PARAMDEFN2", and so on.

For example:

```

Parameters=["__UID__":"Type:String,Direction:OUT,ColName:USERNAME",
"authType":"Type:String,Direction:OUT,ColName:PASSWORD,ColQuery:\\"DECODE(PASSWORD, 'EXTERNAL', 'EXTERNAL', 'GLOBAL', 'GLOBAL', 'PASSWORD')\\""]

```

A parameter can have the following attributes:

- **Type** is the type of the parameter.
- **Direction** is the flow of data from the query to or from the parameter. It can have a value of IN, OUT, or INOUT.
- **ColName** is the column name in the target system corresponding to the parameter in the query.
- **ColQuery** is the query used to fetch values for the corresponding query parameter.
- **ExtensionJoin** (optional) refers to the operator, represented by **EXTENSIONJOIN**, used to join the main query with query extensions.

For example: ExtensionJoin=","

- `ExtensionSeparator` (optional) refers to the delimiter between query extensions, represented by `EXTENSIONSEPARATOR`.
For example: `ExtensionSeparator=", "`
- `QueryExtensions` (optional) refers to the extensions that must be appended to the main query, represented by `EXTENSION1`, `EXTENSION2`, and so on.
For example: `QueryExtensions=["SEARCH_USER_ROLE", "SEARCH_USER_PRIVILEGE"]`

During a reconciliation operation, the connector combines all these components to the following query:

```
QUERY PARAM1, PARAM2... [EXTENSIONJOIN [EXTENSION1 EXTENSIONSEPARATOR
EXTENSION2 EXTENSIONSEPARATOR...]]
```

For example:

```
SELECT {__UID__}, {authType}, {externalname}, {tablespace}, {status},
{tempTableSpace}, {profile}, {defaultQuota}, {tmpQuota}, {lastModified} FROM
DBA_USERS dba {filter}, SEARCH_USER_ROLE, SEARCH_USER_PRIVILEGE
```

4.7.1.4 Syntax of List of Values Queries for Oracle Database

If a search query is performed on account types, such as User Name, then the query is considered as a reconciliation query. If a search query is performed on any other object, then the query is considered as a list of values query.

The following is the syntax of the list of values queries used for lookup field synchronization:

```
OBJECTTYPE = "QUERY"
```

For example:

```
__PROFILE__="SELECT DISTINCT profile FROM dba_profiles"
```

In this syntax:

- `OBJECTTYPE` refers to the lookup field attribute.
For example: `__PROFILE__`
- `QUERY` refers to the query used for fetching a lookup field attribute.
For example: `SELECT DISTINCT profile FROM dba_profiles`

The list of values queries return values that are used as lookup field entries. By default, the connector includes dedicated scheduled job for each lookup definition. To use a custom lookup definition, you must add custom fields in the query file.

4.7.1.5 Guidelines for Configuring Search Queries Used in Reconciliation from Oracle DatabaseS

The following are guidelines that you must apply while modifying or creating queries for reconciliation:

- By adding or removing a column from the `SELECT` clause of a reconciliation query, you add or remove an attribute from the list of target system attributes for reconciliation. To enable the connector to process a change (addition or removal) in the list of reconciled attributes, you must make corresponding changes in the provisioning part of the connector.

If there are any read-only attributes, then you must disable updates to the read-only attributes in the respective process forms.

- In the query properties file, you must not change the names of the predefined queries.
- Some of the predefined queries use inner queries. If you add or remove a column from the outer query, you must make corresponding changes in the inner queries.
- You cannot remove columns corresponding to the User Name resource object attribute.
- You must ensure that the following condition included in the Parameters list is not removed:

```
"lastModified": "Type:long,Direction:IN,ColQuery:\"((CREATED -  
TO_DATE('01011970', 'ddmmyyyy')) *24*60*60*1000)\""]
```

This condition is used to determine if a target system record was added or updated after the time-stamp stored in the Incremental Recon Attribute scheduled job attribute.

- You must ensure that formats for date literals are specified by the use of the TO_DATE function. For example, instead of specifying a date value as '31-Dec-4712' use TO_DATE('31-Dec-4712', 'DD-Mon-YYYY').
- When you add or remove columns from the SELECT clause of the queries in the properties file, then you must update the attribute mapping lookup definition that holds mappings between child attributes and the target system column names. In addition, you must update other OIM objects.
- Before you modify or add a query in the Search.queries file, you must run the query by using any standard database client to ensure that the query produces the required results when it is run against the target system database.

4.7.2 Configuring Queries to Add Support for Custom Parameters and Lookup Fields for Oracle Database

The connector uses preconfigured queries for connector operations such as create, delete, and search. You can add custom parameters and lookup definition fields as per your requirements.

The procedure to add a parameter or a lookup definition field to a query file is discussed in the following sections:

- [Updating the Query Files for Oracle Database](#)
- [Configuring Oracle Identity Manager](#)

4.7.2.1 Updating the Query Files for Oracle Database

To update the query files:

1. If the connector is already installed, run the Oracle Identity Manager Download JARs utility to download the connector bundle JAR file from the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

```
OIM_HOME/server/bin/DownloadJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/DownloadJars.sh
```

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

2. Copy the bundle JAR file in a temporary directory.

Sample JAR file: `bundle/org.identityconnectors.dbum-1.0.1116.jar`

Sample temporary directory: `c:\temp`

3. Run the following command to extract the connector bundle JAR file:

```
jar -xvf org.identityconnectors.dbum-1.0.1116.jar
```

 **Note:**

You can also run the WinZip or WinRAR utility to extract the contents from the JAR file.

4. Delete the bundle JAR file in the temporary directory.
5. Update the value of **ConnectorBundle-Version** in the manifest file, `META-INF/MANIFEST.MF`, to a new value.

For example:

```
ConnectorBundle-Version: 1.0.1117
```

6. Depending on your requirement, update the query files with new parameters as per the query syntax described in [About the Queries for Oracle Database](#).

For example, if you want to add a new parameter, `tmpQuota`, to the `CREATE_USER` provisioning query:

- a. Open the provisioning query file in a text editor.

Sample query file:

```
c:\temp\bundle\org.identityconnectors.dbum-1.0.1116\scripts\oracle\Provisioning.queries
```

- b. Add the parameter, `tmpQuota`, to the `CREATE_USER` query.

The following is a sample updated query:

```
CREATE_USER {  
  Query="CREATE USER {__NAME__} IDENTIFIED BY {__PASSWORD__} TEMPORARY QUOTA
```

```

{tmpQuota} ON {tempTableSpace}"
  QueryType="SQL"
  Parameters=["__NAME__": "Type:String, TAGS:DOUBLEQUOTES",
"__PASSWORD__": "Type:GuardedString, TAGS:DOUBLEQUOTES",
"tmpQuota": "Type:String",
"tempTableSpace": "Type:String, Tags:EXCLUDE_VALIDATION"]

QueryExtensions=["TABLESPACE_QUERY", "TEMP_TABLESPACE_QUERY", "PROFILE_QUERY",
"DEFAULTS_QUOTA_QUERY", "TEMPTS_QUOTA_QUERY"]
}

```

- c. Save and close the query file.
7. Create a new bundle JAR file that contains the updated manifest file and the provisioning query file as follows:

- a. Open the command prompt and navigate to the temporary directory:

```
c:\temp
```

- b. Run the following command:

```
jar -cvfm org.identityconnectors.dbum-1.0.1117.jar *
```

The new connector bundle JAR name contains the new bundle version.

8. In the case of a remote connector server, copy the new bundle JAR file in the bundles directory of the remote connector server, instead of posting the JAR file to the Oracle Identity Manager database. Skip to Step 10.
9. Run the Oracle Identity Manager Update JARs utility to update the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

Note:

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

If you have installed both the Oracle and MSSQL connectors on the same Oracle Identity Manager, then ensure that all third-party JAR files are part of the `/lib` directory in the connector bundle JAR file.

For Microsoft Windows:

```
OIM_HOME/server/bin/UpdateJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UpdateJars.sh
```

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

10. Update the configuration lookup with the new bundle version.

For example, you can update the `Lookup.DBUM.Oracle.Configuration` lookup definition.

4.7.2.2 Configuring Oracle Identity Manager

You can skip this procedure if the parameter you added already exists as a default form field in Oracle Identity Manager.

To configure Oracle Identity Manager for adding a parameter:

1. Log into Oracle Identity Manager Design Console.
2. Create a new version of the process form:
 - a. Expand **Development Tools**.
 - b. Double-click **Form Designer**.
 - c. Search for and open the **UD_DB_ORA_U** process form.
 - d. Click **Create New Version**.

On the Create a new version dialog box, enter a new version in the Label field, and then click the save icon.
3. Add the new field on the process form.
 - a. Click **Add**.

A field is added to the list. Enter the details of the field.

For example, if you are adding the tmpQuota field, enter `UD_DB_ORA_U_TMPQUOTA1` in the **Name** field and then enter the rest of the details of this field.
 - b. Click the save icon and then click **Make Version Active**.
4. If you are using Oracle Identity Manager release 11.1.2.x or later, then all changes made to the Form Designer of the Design Console must be done in a new UI form as follows:
 - a. Log in to Oracle Identity System Administration.
 - b. Create and active a sandbox.
 - c. Create a new UI form to view the newly added field along with the rest of the fields. See *Creating Forms By Using the Form Designer in Oracle Fusion Middleware Administering Oracle Identity Manager*.
 - d. Associate the newly created UI form with the application instance of your target system. To do so, open the existing application instance for your resource, from the Form field, select the form (created in Step 4.c), and then save the application instance.
 - e. Publish the sandbox as described in *Publishing a Sandbox of Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.
5. Click **Add** and enter the task name, for example, `Custom Attribute 1 Updated`, and the task description.
6. Create an entry for the field in the lookup definition for provisioning as follows:
 - a. Expand **Administration**.
 - b. Double-click **Lookup Definition**.
 - c. Search for and open the **Lookup.DBUM.Oracle.UM.ProvAttrMap** lookup definition.
 - d. Click **Add** and enter the Code Key and Decode values for the field.

The Code Key value must be the form field name. The Decode value must be the attribute name on the target system.

For example, enter `Temporary Quota` in the **Code Key** field and then enter `tmpQuota` in the **Decode** field.

- e. Click the save icon.
7. Create a process task to update the new field `Temporary Quota` as follows:
 - a. Expand **Process Management**.
 - b. Double-click **Process Definition** and open the **Oracle DB User** process definition.
 - c. Click **Add** and enter the task name, for example, `Temporary Quota Updated`, and the task description.
 - d. In the Task Properties section, select **Conditional** and **Allow Multiple Instances** fields and click the save icon.
 - e. Select the **adpORAUPDATEWITHREF** adapter, click the save icon, and then click **OK** in the message that is displayed.
 - f. To map the adapter variables listed in this table, select the adapter, click **Map**, and then specify the data given in the following table:

Variable Name	Data Type	Map To	Qualifier	Literal Value
Adapter return value	Object	Response code	NA	NA
attrName	String	Literal	String	Temporary Quota
ITResField	String	Literal	String	UD_DB_ORA_U_ITRES
newVal	String	Process Data	tmpQuota	NA
objectType	String	Literal	String	User
oldValue	String	Process Data	tmpQuota	NA
Note: The old value check box must be selected.				
proclInstance	Long	Process Data	Process Instance	NA

- g. On the Responses tab, click **Add** to add the following response codes:

Code Name	Description	Status
ERROR	Error occurred	R
UNKNOWN	An unknown response was received	R
SUCCESS	Operation completed	C

- h. Click the save icon and then close the dialog box.

4.7.3 About Configuring the Connector for Multiple Installations of Oracle Database

You might want to configure the connector for multiple installations of the target system. The following example illustrates this requirement:

The London and New York offices of Example Multinational Inc. have their own installations of the target system. The company has recently installed Oracle Identity Manager, and they want to configure Oracle Identity Manager to link all the installations of the target system.

To meet the requirement posed by such a scenario, you can create copies of connector objects, such as the IT resource and resource object.

The decision to create a copy of a connector object might be based on a requirement. For example, an IT resource can hold connection information for one target system installation. Therefore, it is mandatory to create a copy of the IT resource for each target system installation.

With some other connector objects, you do not need to create copies at all. For example, a single attribute-mapping lookup definition can be used for all installations of the target system.

All connector objects are linked. For example, a scheduled job holds the name of the IT resource. Similarly, the IT resource for a target system such as Oracle Database holds the name of the configuration lookup definition, `Lookup.DBUM.Oracle.Configuration`. If you create a copy of an object, then you must specify the name of the copy in associated connector objects.

 **Note:**

- To reconcile data from a particular target system installation, specify the name of the IT resource for that target system installation as the value of the scheduled job attribute that holds the IT resource name. For example, you enter the name of the IT resource as the value of the IT resource attribute of the scheduled job that you run.
- When you use Identity Self Service to perform provisioning, you can specify the IT resource corresponding to the target system installation to which you want to provision the user.

[Table 4-19](#) lists associations between connector objects whose copies can be created and the other objects that reference these objects. When you create a copy of a connector object, use this information to change the associations of that object with other objects.

 **Note:**

- On a particular Oracle Identity Manager installation, if you create a copy of a connector object, then you must set a unique name for it.
- If you are using Oracle Identity Manager release 11.1.2.x or later, then in addition to the procedure described in this section, you must create an application instance for each IT resource. See [Configuring Oracle Identity Manager Release 11.1.2 or Later](#) for information on creating an application instance.

Table 4-19 Connector Objects and Their Associations

Connector Object	Name	Referenced By	Comments on Creating a Copy
IT resource	Oracle	<ul style="list-style-type: none"> UD_DB_ORA_U (process form) Scheduled tasks 	Create a copy of the IT resource with a different name.
Resource object	Oracle DB User Oracle DB Trusted	All connector operations	<p>It is optional to create a copy of the resource object. If you are reconciling the same set of attributes from all installations of the target system, then you need not create a copy of the resource object.</p> <p>Note: Create copies of the resource object only if there are differences in attributes between the various installations of the target system.</p>
Scheduled Jobs	There are many scheduled jobs for different purposes.	NA	You can use the scheduled jobs with the same names. However, you must update the values of the parameters depending on the target system you want to use.
Process definition	Oracle DB User	NA	<p>It is optional to create a copy of the process definition. If you are reconciling or provisioning the same set of attributes from all installations of the target system, then you need not create a copy of the process definition.</p> <p>Note: Create copies of the process form only if there are differences in attributes between the various installations of the target system.</p>
Process form	UD_DB_ORA_U	Oracle DB User (Process definition)	<p>It is optional to create a copy of the process form. If you are provisioning the same set of attributes from all installations of the target system, then you need not create a copy of the process definition.</p> <p>Note: Create copies of the process form only if there are differences in attributes between the various installations of the target system.</p>
Child process form	<ul style="list-style-type: none"> UD_DB_OR_A_R UD_DB_OR_A_P 	<ul style="list-style-type: none"> Oracle DB User (Process definition) UD_DB_ORA_U (Process form) 	It is optional to create a copy of the child process form. If you are provisioning a new set of child data, then you need to create a copy of the child and parent process forms. Then, assign the newly created child process form to the newly created parent process form.

Table 4-19 (Cont.) Connector Objects and Their Associations

Connector Object Name	Referenced By	Comments on Creating a Copy
Configuration lookup definition for a target system configured as a target resource	Lookup.DBUM.Oracle.Configuration	Oracle DB (IT resource)
Configuration lookup definition for a target system configured as a trusted source	Lookup.DBUM.Oracle.Configuration.Trusted	Oracle DB (IT resource)
Resource object attributes mapping lookup definition (for target resource)	Lookup.DBUM.Oracle.UM.ReconAttrMap	NA

Table 4-19 (Cont.) Connector Objects and Their Associations

Connector Object	Name	Referenced By	Comments on Creating a Copy
Configuration lookup definition for a target system configured as a trusted source	Lookup.DBUM.Oracle.UM.ReconAttrMap.Trusted	Oracle DB (IT resource)	<p>It is optional to create a copy of the configuration lookup definition. If you are reconciling the same set of attributes in all installations of the target system (configured as a trusted source), then you need not create a copy of the configuration lookup definition.</p> <p>Note: Create copies of the configuration lookup definition for trusted source only if there are differences in attributes between the various installations of the target system and you have created a new process form.</p>

4.7.4 About Configuring the Connector for Multiple Trusted Source Reconciliation from Oracle Database

Note:

This connector supports multiple trusted source reconciliation.

This section describes an optional procedure. Perform this procedure only if you want to configure the connector for multiple trusted source reconciliation.

The following are examples of scenarios in which there is more than one trusted source for user data in an organization:

- One of the target systems is a trusted source for data about users. The second target system is a trusted source for data about contractors. The third target system is a trusted source for data about interns.
- One target system holds the data of some of the identity fields that constitute an OIM User. Two other systems hold data for the remaining identity fields. In other words, to create an OIM User, data from all three systems would need to be reconciled.

If the operating environment of your organization is similar to that described in either one of these scenarios, then this connector enables you to use the target system as one of the trusted sources of person data in your organization.

See *Managing Reconciliation in Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed information about multiple trusted source reconciliation.

4.7.5 Configuring Validation of Data During Reconciliation and Provisioning for Oracle Database

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

To configure validation of data:

1. Write code that implements the required validation logic in a Java class with a fully qualified domain name (FQDN), such as `org.identityconnectors.dbum.extension.DBUMValidator`.

This validation class must implement the `validate` method. The following sample validation class checks if the value in the First Name attribute contains the number sign (#):

```
package com.validationexample;

import java.util.HashMap;

public class MyValidator {
    public boolean validate(HashMap hmUserDetails, HashMap hmEntitlementDetails,
        String sField) throws ConnectorException {

        /* You must write code to validate attributes. Parent
         * data values can be fetched by using hmUserDetails.get(field)
         * For child data values, loop through the
         * ArrayList/Vector fetched by hmEntitlementDetails.get("Child
Table")

         * Depending on the outcome of the validation operation,
         * the code must return true or false.
         */

        /*
         * In this sample code, the value "false" is returned if the field
         * contains the number sign (#). Otherwise, the value "true" is
         * returned.
         */
        boolean valid = true;
        String sFirstName = (String) hmUserDetails.get(sField);
        for (int i = 0; i < sFirstName.length(); i++) {
            if (sFirstName.charAt(i) == '#') {
                valid = false;
                break;
            }
        }
        return valid;
    }
}
```

2. Log in to the Design Console.
3. Search for and open one of the lookup definitions (or create a new lookup) listed in [Lookup Definition for Validation of Data in Oracle Database](#).

For example, **Lookup.DBUM.Oracle.UM.ProvValidations**.

 **Note:**

If you cannot find these lookup definitions, create new lookup definitions.

4. In the **Code Key** column, enter the resource object field name that you want to validate. For example, `Username`.
5. In the **Decode** column, enter the class name. For example, `org.identityconnectors.dbum.extension.DBUMValidator`.
6. Save the changes to the lookup definition.
7. Search for and open the configuration lookup definition for the target system you use.

For example, **Lookup.DBUM.Oracle.UM.Configuration**.

8. In the **Code Key** column, enter one of the following entries:
 - To configure validation of data for reconciliation:
`Recon Validation Lookup`
 - To configure validation of data for provisioning:
`Provisioning Validation Lookup`
9. In the **Decode** column, enter the name of the lookup you updated or created in step 3.

For example, **Lookup.DBUM.Oracle.UM.ProvValidations**.

10. Save the changes to the lookup definition.
11. Create a JAR with the class and upload it to the Oracle Identity Manager database as follows:

Run the Oracle Identity Manager Upload JARs utility to post the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

```
OIM_HOME/server/bin/UploadJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UploadJars.sh
```

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

12. Run the PurgeCache utility to clear content related to request datasets from the server cache.
13. Perform reconciliation or provisioning to verify validation for the field, for example, Username.

4.7.6 Configuring Transformation of Data During User Reconciliation for Oracle Database

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

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

1. Write code that implements the required transformation logic in a Java class with a fully qualified domain name (FQDN), such as `org.identityconnectors.dbum.extension.DBUMTransformation`.

This transformation class must implement the transform method. The following sample transformation class modifies the Username attribute by using values fetched from the `__NAME__` attribute of the target system:

```
package com.transformationexample;

import java.util.HashMap;

public class MyTransformer {
    public Object transform(HashMap hmUserDetails, HashMap hmEntitlementDetails,
String sField) throws ConnectorException {
        /*
         * You must write code to transform the attributes.
         * Parent data attribute values can be fetched by
         * using hmUserDetails.get("Field Name").
         * To fetch child data values, loop through the
         * ArrayList/Vector fetched by hmEntitlementDetails.get("Child
Table")
         * Return the transformed attribute.
         */
        String sUserName = (String) hmUserDetails.get("__NAME__");
        return sUserName + "@example.com";
    }
}
```

2. Log in to the Design Console.
3. Search for and open one of the lookup definitions (or create a new lookup) listed in [Lookup Definitions for Transformation of Data in Oracle Database](#).

For example, **Lookup.DBUM.Oracle.UM.ReconTransformations**.

Note:

If you cannot find these lookup definitions, create new lookup definitions.

4. In the **Code Key** column, enter the resource object field name you want to transform. For example, `Username`.
5. In the **Decode** column, enter the class name. For example, `org.identityconnectors.dbum.extension.DBUMTransformation`.
6. Save the changes to the lookup definition.
7. Search for and open the **Lookup.DBUM.Oracle.UM.Configuration** lookup definition.
8. In the **Code Key** column, enter `Recon Transformation Lookup`.
9. In the **Decode** column, enter the name of the lookup you updated or created in step 3.
For example, **Lookup.DBUM.Oracle.UM.ReconTransformations**.
For trusted mode, use **Lookup.DBUM.Oracle.UM.ReconTransformations.Trusted**.
10. Save the changes to the lookup definition.
11. Create a JAR with the class and upload it to the Oracle Identity Manager database as follows:
Run the Oracle Identity Manager Upload JARs utility to post the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

```
OIM_HOME/server/bin/UploadJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UploadJars.sh
```

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

12. Run the `PurgeCache` utility to clear content related to request datasets from the server cache.
13. Perform reconciliation to verify transformation of the field, for example, `SimpleDisplayName`.

4.7.7 Configuring Resource Exclusion Lists for Oracle Database

You can specify a list of accounts that must be excluded from reconciliation and provisioning operations. Accounts whose user IDs you specify in the exclusion list are not affected by reconciliation and provisioning operations.

In one of the lookup definitions for exclusion lists, enter the user IDs of target system accounts for which you do not want to perform provisioning and reconciliation operations. See [Lookup Definitions for Exclusion Lists for Oracle Database](#) for information about the lookup definitions and the format of the entries in these lookups.

To add entries in the lookup for exclusions during provisioning and reconciliation operations for Oracle Database:

1. On the Design Console, expand **Administration** and then double-click **Lookup Definition**.
2. Search for and open the **Lookup.DBUM.Oracle.UM.ExclusionList** lookup definition.
3. Click **Add**.
4. In the Code Key column, enter the resource object field name on which the exclusion list is applied. In the Decode column, enter the corresponding ID of the record to exclude.

For example, if you do not want to provision users with the user ID User001, then you must populate the lookup definition with the following values:

Code Key	Decode
User Name	User001

 **Note:**

If you want to specify a list of accounts that must be excluded during reconciliation or provisioning, the code key value being specified here must be exactly as the corresponding code key value in the `Lookup.DBUM.Oracle.UM.ReconAttrMap` lookup definition, or in the `Lookup.DBUM.Oracle.UM.ProvAttrMap` lookup definition, respectively.

5. If there is more than one user ID to exclude, then in the decode column, enter a list of all user IDs to exclude. Note that each User ID must be separated by a vertical bar (|).

For example, if you do not want to provision users with user IDs User001, User002, and User088 then you must populate the lookup definition with the following values:

Code Key	Decode
User Name	User001 User002 User088

You can also perform pattern matching to exclude user accounts. You can specify regular expressions supported by the representation in the `java.util.regex.Pattern` class.

 **See Also:**

For information about the supported patterns, visit <http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html>

For example, if you do not want to provision users matching any of the user IDs User001, User002, and User088, then you must populate the lookup definition with the following values:

Code Key	Decode
User Name[PATTERN]	User001 User002 User088

If you do not want to provision users whose user IDs start with 00012, then you must populate the lookup definition with the following values:

Code Key	Decode
User Name[PATTERN]	00012*

- Click the save icon.

4.7.8 Setting Up Action Scripts for Oracle Database

Learn about action scripts and how to configure them to run before or after the create, update, or delete an account provisioning operations.

This section provides information about the following topics:

- [About Action Scripts for Oracle Database](#)
- [Configuring Action Scripts for Oracle Database](#)

4.7.8.1 About Action Scripts for Oracle Database

Actions are scripts that you can configure to run before or after the create, update, or delete an account provisioning operations. For example, you could configure a script to run before every user creation. In another scenario, suppose you have a table called AUDIT_USERLOG where you want to log user creation activities performed only by the connector. Then, you could create and use after create script for adding data to this table after create operation.

Note:

To configure a before or after action, your connector must support running scripts. An exception is Groovy (with target set to **Connector**), which the Identity Connector Framework (ICF) supports by default for all converged connectors.

Every connector should specify which scripting language and which target it supports. This connector supports the following script:

- shell:** shell script
- target:** Connector

The target refers to the location where the script is executed. In this case, the script is executed on the same computer (JVM or .NET Runtime) where the connector is

deployed. For example, if you deploy the connector on the connector server, the script will be executed on that computer.

That is, if you are using a local framework, the script runs in your JVM. If you are connected to a remote framework, the script runs in the remote JVM or .NET Runtime.

4.7.8.2 Configuring Action Scripts for Oracle Database

To configure the action:

1. Log in to the Design Console.
2. Search for and open the **Lookup.DBUM.Oracle.UM.Configuration** lookup definition.
3. Add the following new values:
 - **Code Key:** Before Create Action Language
 - **Decode:** Enter the scripting language of the script you want to execute
 - **Sample values:** SQL or STOREDPROC

4. Add these new values:

- **Code Key:** Before Create Action File
- **Decode:** Enter the full path to the file containing the script to be executed (Oracle Identity Manager must be able to access this file.)
- **Example:** /home/scripts/testscript.sql

This script may have a query as follows:

```
INSERT INTO AUDIT_USERLOG VALUES ({{__NAME__}}, CURRENT_TIMESTAMP))
```

5. Add these new values:

- **Code Key:** Before Create Action Target
- **Decode:** Connector

6. Save the lookup definition.

Now, this action will be executed every time you create a user. You must configure these three values for each action you want to execute.

5

Using and Extending the Connector for MySQL

You can use the Database User Management Connector for MySQL for performing reconciliation and provisioning operations after configuring it to meet your requirements. You can also extend the functionality of the connector to address your specific business requirements.

This chapter contains the following topics:



Note:

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

For Oracle Identity Manager hosted on a Microsoft Windows computer, if you have a previously installed connector, then you must extract the connector bundle zip file again before installing a new connector.

- [Configuring Secure Communication Between MySQL and Oracle Identity Manager](#)
- [Guidelines on Determining Values for the JDBC URL and Connection Properties Parameters for MySQL](#)
- [Lookup Definitions for MySQL](#)
- [Scheduled Jobs for MySQL](#)
- [Reconciliation from MySQL](#)
- [Provisioning for MySQL](#)
- [Extending the Connector for MySQL](#)

5.1 Configuring Secure Communication Between MySQL and Oracle Identity Manager



Note:

It is recommended that you perform the procedure described in this section to secure communication between the target system and Oracle Identity Manager.

Perform the following steps to configure secure communication between MySQL and Oracle Identity Manager:

1. See MySQL documentation for information about enabling SSL communication between MySQL and a client system. In this context, the client is Oracle Identity Manager.
2. Export the certificate on the MySQL host computer.
3. Restart the MySQL database service by using the certificate exported in the preceding step. See MySQL documentation for information on restarting the database service.
4. Copy the `ca-cert.pem` and `client-cert.pem` certificates to the Oracle Identity Manager host computer.
5. Import the certificates into the JVM truststore of the application server on which Oracle Identity Manager is running.

To import the certificates into the truststore, run the following command for each certificate:

```
keytool -import -file FILE_LOCATION -keystore TRUSTSTORE_LOCATION -storepass TRUSTSTORE_PASSWORD -trustcacerts -alias ALIAS
```

In this command:

- Replace *FILE_LOCATION* with the full path and name of the certificate file.
- Replace *ALIAS* with an alias for the certificate.
- Replace *TRUSTSTORE_PASSWORD* with a password for the truststore.
- Replace *TRUSTSTORE_LOCATION* with one of the truststore paths from [Table 5-1](#). This table shows the location of the truststore for each of the supported application servers.

 **Note:**

In an Oracle Identity Manager cluster, import the file into the truststore on each node of the cluster.

Table 5-1 Truststore Locations on Supported Application Servers

Application Server	Truststore Location
Oracle WebLogic Server	<ul style="list-style-type: none"> • If you are using Oracle <code>jrockit_R27.3.1-jdk</code>, then import the certificate into the keystore in the following directory: <code>JROCKIT_HOME/jre/lib/security</code> • If you are using the default Oracle WebLogic Server JDK, then import the certificate into the keystore in following directory: <code>WEBLOGIC_HOME/java/jre/lib/security/cacerts</code> • If you are using a JDK other than Oracle <code>jrockit_R27.3.1-jdk</code> or Oracle WebLogic Server JDK, then import the certificate into your keystore at the following directory: <code>JAVA_HOME/jre/lib/security/cacerts</code>

6. To enable secure communication between MySQL and Oracle Identity Manager, set the value of the UseSSL IT resource parameter to `true`. You must provide a value for this parameter while performing the procedure described in [Configuring the IT Resource for the Connector Server](#).

5.2 Guidelines on Determining Values for the JDBC URL and Connection Properties Parameters for MySQL

This section discusses the JDBC URL and Connection Properties parameters. You apply the information in this section while performing the procedure described in [Configuring the IT Resource for the Target System](#).

The following are guidelines on specifying the JDBC URL and Connection Properties parameters:

- **JDBC URL parameter**

Enter the following component of the connection URL as the value of the JDBC URL provider:

```
jdbc:mysql://[SERVER_NAME][:PORT_NUMBER]/[DATABASE_NAME]
```

In this format:

- *SERVER_NAME* is the IP address (not the host name) of the target system host computer.
- *PORT_NUMBER* is the port at which the target system database is listening.
- *DATABASE_NAME* is the name of the database we are connecting.

The following is a sample value for the Database URL parameter:

```
jdbc:mysql://192.168.16.76:3306/information_schema
```

- **Connection Properties parameter**

Enter the following component of the connection URL as the value of the Connection Properties parameter:

```
[, PROPERTY=VALUE[, PROPERTY=VALUE]] . . .
```

In this format:

- *PROPERTY* is the name of one or more database connection properties, such as `applicationName` and `disableStatementPooling`.
- *VALUE* is the value of each database connection property whose name you specify by using the *PROPERTY* placeholder.

 **Note:**

Semicolons must be changed to number signs (#) in the value that you specify.

The following is a sample value for the Connection Properties parameter:

```
databaseName=information_schema#port=3306
```

If you enable SSL communication between MySQL and Oracle Identity Manager, then:

- Append the following value to the value in the Connection Properties parameter of the IT resource:

```
useSSL=true#requireSSL=true
```

For example, suppose the following is the existing value for the Connection Properties parameter:

```
databaseName=information_schema#port=3306
```

Now, if you enable SSL communication between MySQL and Oracle Identity Manager, then the value of the Connection Properties parameter must be as follows:

```
databaseName=information_schema#port=3306#useSSL=true#requireSSL=true
```

5.3 Lookup Definitions for MySQL

This section contains the following information:

- [Lookup Definitions Synchronized with MySQL](#)
- [Lookup Definitions for Configurations for MySQL](#)
- [Lookup Definitions for Attribute Mappings for MySQL](#)
- [Lookup Definitions for Exclusion Lists for MySQL](#)
- [Lookup Definitions for Transformation of Data in MySQL](#)
- [Lookup Definition for Validation of Data in MySQL](#)

5.3.1 Lookup Definitions Synchronized with MySQL

During a provisioning operation, you use a lookup field on the process form to specify a single value from a set of values. For example, you use the Privileges lookup field to select a privilege to be assigned to a schema from the list of available privileges. When you deploy the connector, lookup definitions corresponding to the lookup fields on the target system are created in Oracle Identity Manager. Lookup field synchronization involves copying additions or changes made to the target system lookup fields into the lookup definitions in Oracle Identity Manager.

The connector provides predefined SQL queries for fetching values from the target system lookup fields into the lookup definitions in Oracle Identity Manager. These predefined SQL queries are stored in the `LoVSearch.queries` file with in the connector bundle.

After lookup definition synchronization, data is stored in the following format:

- Code Key value: `IT_RESOURCE_KEY~LOOKUP_FIELD_ID`

In this format:

- `IT_RESOURCE_KEY` is the numeric code assigned to each IT resource in Oracle Identity Manager.
- `LOOKUP_FIELD_ID` is the target system code assigned to each lookup field entry.

Sample value: `1~SYS_ADM`

- Decode value: `IT_RESOURCE_NAME~LOOKUP_FIELD_ID`

In this format:

- `IT_RESOURCE_NAME` is the name of the IT resource in Oracle Identity Manager.
- `LOOKUP_FIELD_ID` is the target system code assigned to each lookup field entry.

Sample value: `MySQL_DB~SYS_ADM`

While performing a provisioning operation in Oracle Identity Self Service, you select the IT resource for the target system on which you want to perform the operation. When you perform this action, the lookup definitions on the page are automatically populated with values corresponding to the IT resource (target system installation) that you select. If your environment has multiple installations of the target system, then values corresponding to all IT resources are displayed.

[Table 5-2](#) lists column name of the table in MySQL that is synchronized with its corresponding lookup definition in Oracle Identity Manager.

Table 5-2 Lookup Definitions Synchronized with MySQL

Lookup Definition	Target Column Name
Lookup.DBUM.MySQL.SchemaPrivileges	Privilege

5.3.2 Lookup Definitions for Configurations for MySQL

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

This section provides information about the following lookup definitions

- [Lookup.DBUM.MySQL.Configuration](#)
- [Lookup.DBUM.MySQL.UM.Configuration](#)
- [Lookup.DBUM.MySQL.Configuration.Trusted](#)
- [Lookup.DBUM.MySQL.UM.Configuration.Trusted](#)

5.3.2.1 Lookup.DBUM.MySQL.Configuration

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

Table 5-3 Entries in Lookup.DBUM.MySQL.Configuration

Code Key	Decode Key	Description
Bundle Name	<code>org.identityconnectors.dbum</code>	Name of the connector bundle package Do not modify this entry.
Bundle Version	1.0.1116	Version of the connector bundle class Do not modify this entry.
Connector Name	<code>org.identityconnectors.dbum.DBUMConnector</code>	Name of the connector class Do not modify this entry.

Table 5-3 (Cont.) Entries in Lookup.DBUM.MySQL.Configuration

Code Key	Decode Key	Description
User Configuration Lookup	Lookup.DBUM.MySQL.UM.Configuration	Name of the lookup definition that contains user-specific configuration properties Do not modify this entry.

5.3.2.2 Lookup.DBUM.MySQL.UM.Configuration

The Lookup.DBUM.MySQL.UM.Configuration lookup definition holds user-specific connector configuration entries that are used during target resource reconciliation and provisioning operations.

Table 5-4 Entries in Lookup.DBUM.MySQL.UM.Configuration

Code Key	Decode Key
Provisioning Attribute Map	Lookup.DBUM.MySQL.UM.ProvAttrMap
Provisioning Exclusion List	Lookup.DBUM.MySQL.UM.ExclusionList
Provisioning Validation Lookup	Lookup.DBUM.MySQL.UM.ProvValidations
Recon Validation Lookup	Lookup.DBUM.MySQL.UM.ReconValidations
Recon Attribute Map	Lookup.DBUM.MySQL.UM.ReconAttrMap
Recon Exclusion List	Lookup.DBUM.MySQL.UM.ExclusionList
Recon Transformation Lookup	Lookup.DBUM.MySQL.UM.ReconTransformations

5.3.2.3 Lookup.DBUM.MySQL.Configuration.Trusted

The Lookup.DBUM.MySQL.Configuration.Trusted lookup definition holds connector configuration entries that are used during reconciliation and provisioning operations in trusted source mode.

Table 5-5 Entries in Lookup.DBUM.MySQL.Configuration.Trusted

Code Key	Decode Key
Bundle Name	org.identityconnectors.dbum
Bundle Version	1.0.1116
Connector Name	org.identityconnectors.dbum.DBUMConnector
User Configuration Lookup	Lookup.DBUM.MySQL.UM.Configuration.Trusted

5.3.2.4 Lookup.DBUM.MySQL.UM.Configuration.Trusted

The Lookup.DBUM.MySQL.UM.Configuration.Trusted lookup definition holds user-specific connector configuration entries that are used during reconciliation and provisioning operations in trusted source mode.

Table 5-6 Entries in Lookup.DBUM.MySQL.UM.Configuration.Trusted

Code Key	Decode Key
Recon Attribute Defaults	Lookup.DBUM.MySQL.UM.ReconDefaults.Trusted
Recon Attribute Map	Lookup.DBUM.MySQL.UM.ReconAttrMap.Trusted
Recon Validation Lookup	Lookup.DBUM.MySQL.UM.ReconValidations.Trusted
Recon Exclusion List	Lookup.DBUM.MySQL.UM.ExclusionList.Trusted
Recon Transformation Lookup	Lookup.DBUM.MySQL.UM.ReconTransformations.Trusted

5.3.3 Lookup Definitions for Attribute Mappings for MySQL

This section describes the following lookup definitions:

- [Lookup.DBUM.MySQL.UM.ProvAttrMap](#)
- [Lookup.DBUM.MySQL.UM.ReconAttrMap](#)
- [Lookup.DBUM.MySQL.UM.ReconAttrMap.Trusted](#)
- [Lookup.DBUM.MySQL.UM.ReconDefaults.Trusted](#)

5.3.3.1 Lookup.DBUM.MySQL.UM.ProvAttrMap

The Lookup.DBUM.MySQL.UM.ProvAttrMap lookup definition holds user-specific mappings between process form fields (Code Key values) and target system attributes (Decode values) used during provisioning operations.

Table 5-7 Entries in Lookup.DBUM.MySQL.UM.ProvAttrMap

Code Key	Decode Key
Return Id	__UID__
UD_DB_MYS_P~Privilege[LO OKUP]	privileges~DBPrivilege~__NAME__
User Name	__NAME__
User Password	__PASSWORD__

5.3.3.2 Lookup.DBUM.MySQL.UM.ReconAttrMap

The Lookup.DBUM.MySQL.UM.ReconAttrMap lookup definition holds user-specific mappings between reconciliation attribute names as specified in the resource object (Code Key values) and target system attributes (Decode values) used during reconciliation operations.

Table 5-8 Entries in Lookup.DBUM.MySQL.UM.ReconAttrMap

Code Key	Decode Key
Privilege List~Privilege Name[LOOKUP]	privileges~DBPrivilege~__NAME__
Return ID	__UID__

Table 5-8 (Cont.) Entries in Lookup.DBUM.MySQL.UM.ReconAttrMap

Code Key	Decode Key
User Name	__UID__

5.3.3.3 Lookup.DBUM.MySQL.UM.ReconAttrMap.Trusted

The Lookup.DBUM.MySQL.UM.ReconAttrMap.Trusted lookup definition holds user-specific mappings between reconciliation attribute names as specified in the resource object (Code Key values) and target system attributes (Decode values) used during reconciliation operations in trusted source mode.

Table 5-9 Entries in Lookup.DBUM.MySQL.UM.ReconAttrMap.Trusted

Code Key	Decode Key
First Name	__NAME__
User ID	__UID__

5.3.3.4 Lookup.DBUM.MySQL.UM.ReconDefaults.Trusted

This lookup definition contains the default values for the Oracle Identity Manager user attributes. You can change these values as per your requirements.

For example, if you want the users reconciled from a trusted source to be part of the MyORG organization, then map the lookup definition entry as follows:

Code Key = Organization Name

Decode = MyORG (instead of Xellerate Users)

Table 5-10 Entries in Lookup.DBUM.MySQL.UM.ReconDefaults.Trusted

Code Key	Decode Key
Empl Type	Full-Time
Organization Name	Xellerate Users
Status	Active
User Type	End-User

5.3.4 Lookup Definitions for Exclusion Lists for MySQL

This section describes the lookup definitions that hold resources for which you do not want to perform provisioning and reconciliation operations. Exclusions can be applied to any attribute in the process form or reconciliation profile. The Code Key value must be one of the Code Key values in Lookup.DBUM.MySQL.UM.ReconAttrMap or Lookup.DBUM.MySQL.UM.ProvAttrMap lookup definitions.

Depending on how the target system is configured, you can use one of the following lookups:

- For target resource mode: Lookup.DBUM.MySQL.UM.ExclusionList
By default, this lookup definition has the following entry:

Code Key	Decode
User Name	root

- For trusted source mode: Lookup.DBUM.MySQL.UM.ExclusionList.Trusted
By default, this lookup definition has the following entry:

Code Key	Decode
User ID	root

The following is the format of the values stored in these lookups:

Code Key	Decode	Sample Values
User Name	User ID of a user	Code Key: User Name Decode: User001
User Name with the [PATTERN] suffix	A regular expression supported by the representation in the <code>java.util.regex.Pattern</code> class	Code Key: User Name[PATTERN] To exclude users matching any of the user ID 's User001, User002, User088, then: Decode: User001 User002 User088 To exclude users whose user ID 's start with 00012, then: Decode: 00012* See Also: For information about the supported patterns, visit http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html

[Configuring Resource Exclusion Lists for MySQL](#) describes the procedure to add entries in these lookup definitions.

5.3.5 Lookup Definitions for Transformation of Data in MySQL

This section describes the lookup definitions that hold resources for which you want to enable transformation of data during reconciliation operations.

Depending on how the target system is configured, use one of the following lookup definitions:

- For target resource mode: Lookup.DBUM.MySQL.UM.ReconTransformations
- For trusted source mode: Lookup.DBUM.MySQL.UM.ReconTransformations.Trusted

[Configuring Transformation of Data During User Reconciliation for MySQL](#) describes the procedure to add entries in these lookup definitions.

5.3.6 Lookup Definition for Validation of Data in MySQL

You can use the Lookup.DBUM.MySQL.UM.ProvValidations lookup to configure validation of data during provisioning operations.

[Configuring Validation of Data During Reconciliation and Provisioning for MySQL](#) describes the procedure to add entries in this lookup definition.

5.4 Scheduled Jobs for MySQL

When you run the Connector Installer or import the connector XML file, the scheduled jobs are automatically created in Oracle Identity Manager.

This section describes the following topics:

- [Scheduled Job for Lookup Field Synchronization for MySQL](#)
- [Attributes for Scheduled Jobs for MySQL](#)
- [Configuring Scheduled Jobs for MySQL](#)

5.4.1 Scheduled Job for Lookup Field Synchronization for MySQL

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

The DBUM MySQL Privilege Type Lookup Reconciliation scheduled job is used for lookup field synchronization.

You must specify values for the attributes of this scheduled job. [Table 5-11](#) describes the attributes of this scheduled job. The procedure to configure scheduled job is described later in the guide.

Table 5-11 Attributes of the Scheduled Job for Lookup Field Synchronization

Attribute	Description
Code Key Attribute	Enter the name of the connector or target system attribute that is used to populate the Code Key column of the lookup definition (specified as the value of the Lookup Name attribute). Sample value: <code>__NAME__</code> Note: Do not change the value of this attribute.
Decode Attribute	Enter the name of the connector or target system attribute that is used to populate the Decode column of the lookup definition (specified as the value of the Lookup Name attribute). Sample value: <code>__NAME__</code>
IT Resource Name	Enter the name of the IT resource for the target system installation from which you want to reconcile user records. Default value: <code>MySQL DB</code>
Lookup Name	This attribute holds the name of the lookup definition that maps each lookup definition with the data source from which values must be fetched. Default value: <code>Lookup.DBUM.MySQL.SchemaPrivileges</code>
Object Type	Enter the type of object whose values must be synchronized. Default value: <code>__PRIVILEGES__</code> Note: Do not change the value of this attribute.
Resource Object Name	Enter the name of the resource object that is used for reconciliation. Default value: <code>MySQL DB User</code>

5.4.2 Attributes for Scheduled Jobs for MySQL

The following scheduled jobs are used to reconcile user data in the target resource (account management) mode of the connector:

- DBUM MySQL User Target Reconciliation
- DBUM MySQL Delete User Target Reconciliation

The following scheduled jobs are used to reconcile user data in the trusted source (identity management) mode of the connector:

- DBUM MySQL User Trusted Reconciliation
- DBUM MySQL Delete User Trusted Reconciliation

[Table 5-12](#) describes the attributes of the scheduled jobs for user operations.

Table 5-12 Attributes of the Scheduled Jobs for Reconciliation

Attribute	Description
Batch Size	Value for running the scheduled job in batch mode. By default, this value is empty.
Filter	Expression for filtering records that must be reconciled by the scheduled job By default, the value of this attribute is empty. Sample value: <code>equalTo('__UID__', 'SEPT12USER1')</code> See Performing Limited Reconciliation from MySQL for the syntax of this expression.
IT Resource Name	Name of the IT resource for the target system installation from which you want to reconcile user records For DBUM MySQL User Target Reconciliation: <code>MySQL DB</code> For DBUM MySQL User Trusted Reconciliation, enter the name of the IT resource created for trusted source mode.
Object Type	Type of object you want to reconcile Default value: <code>User</code>
Resource Object Name	Name of the resource object that is used for reconciliation For DBUM MySQL User Target Reconciliation: <code>MySQL DB User</code> For DBUM MySQL User Trusted Reconciliation: <code>MySQL DB Trusted</code>
Scheduled Task Name	Name of the scheduled job Note: For the scheduled job included with this connector, you must not change the value of this attribute. However, if you create a copy of the task, then you can enter the unique name for that scheduled job as the value of this attribute.

[Table 5-13](#) describes the attributes of the scheduled jobs for delete operations.

Table 5-13 Attributes of the Scheduled Jobs for Delete Operations

Attribute	Description
IT Resource Name	Name of the IT resource for the target system installation from which you want to reconcile user records For DBUM MySQL Delete User Target Reconciliation: <code>MySQL DB</code> For DBUM MySQL Delete User Trusted Reconciliation, enter the name of the IT resource created for trusted source mode.
Object Type	Type of object you want to reconcile Default value: <code>User</code>
Resource Object Name	Name of the resource object that is used for reconciliation For DBUM MySQL Delete User Target Reconciliation: <code>MySQL DB User</code> For DBUM MySQL Delete User Trusted Reconciliation: <code>MySQL DB Trusted</code>

5.4.3 Configuring Scheduled Jobs for MySQL

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

See [Scheduled Job for Lookup Field Synchronization for MySQL](#) and [Attributes for Scheduled Jobs for MySQL](#) for the scheduled jobs that are part of the connector and for information about their attributes.

To configure a scheduled job:

1. For Oracle Identity Manager release 11.1.1.x:
 - a. Log in to the Administrative and User Console.
 - b. On the Welcome to Oracle Identity Manager Self Service page, click **Advanced** in the upper-right corner of the page.
 - c. On the Welcome to Oracle Identity Manager Advanced Administration page, in the System Management region, click **Search Scheduled Jobs**.
2. If you are using Oracle Identity Manager release 11.1.2.x or later:
 - a. Log in to Oracle Identity System Administration.
 - b. In the left pane, under System Management, click **Scheduler**.
3. Search for and open the scheduled job as follows:
 - a. On the left pane, in the Search field, enter the name of the scheduled job as the search criterion. Alternatively, you can click **Advanced Search** and specify the search criterion.
 - b. In the search results table on the left pane, click the scheduled job in the Job Name column.
4. On the Job Details tab, you can modify the following parameters:

Retries: Enter an integer value in this field. This number represents the number of times the scheduler tries to start the job before assigning the Stopped status to the job.

Schedule Type: Depending on the frequency at which you want the job to run, select the appropriate schedule type.

 **Note:**

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

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

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

 **Note:**

- Attribute values are predefined in the connector XML file that you import. Specify values only for those attributes that you want to change.
- Attributes of the scheduled job are discussed in [Attributes for Scheduled Jobs for MySQL](#).

6. After specifying the attributes, click **Apply** to save the changes.

5.5 Reconciliation from MySQL

As mentioned earlier in this guide, reconciliation involves duplicating in Oracle Identity Manager the creation of and modifications to user accounts on the target system. This section discusses the following topics related to configuring reconciliation:

- [Guidelines on Configuring Reconciliation for MySQL](#)
- [About the Reconciliation Process for MySQL](#)
- [Target System Columns Used in Reconciliation from MySQL](#)
- [Configuring the Target System As a Trusted Source](#)
- [Reconciliation Rules for MySQL](#)
- [Viewing Reconciliation Rules for MySQL](#)
- [Reconciliation Action Rules for MySQL](#)
- [Viewing Reconciliation Action Rules for MySQL](#)
- [Performing Full Reconciliation from MySQL](#)
- [Performing Limited Reconciliation from MySQL](#)
- [Performing Batched Reconciliation from MySQL](#)

5.5.1 Guidelines on Configuring Reconciliation for MySQL

The following are guidelines that you must apply while configuring reconciliation:

- Before a target resource reconciliation run is performed, lookup definitions must be synchronized with the lookup fields of the target system. In other words, the scheduled job for lookup field synchronization must be run before user reconciliation runs.

- After you configure batched reconciliation, if reconciliation fails during a batched reconciliation run, then rerun the scheduled job without changing the values of the task attributes.

5.5.2 About the Reconciliation Process for MySQL

This connector can be configured to perform either trusted source reconciliation or target resource reconciliation.

When you configure the target system as a target resource, the connector enables you to create and manage database accounts for OIM Users through provisioning. In addition, data related to newly created and modified target system accounts can be reconciled and linked with existing OIM Users and provisioned resources.

When you configure the target system as a trusted source, the connector fetches into Oracle Identity Manager, data about newly created target system accounts. This data is used to create OIM Users. See [Configuring the Target System As a Trusted Source](#) for more information.

See Also:

Reconciliation Based on the Object Being Reconciled in *Oracle Fusion Middleware Administering Oracle Identity Manager* for conceptual information about target resource reconciliation and trusted source reconciliation.

The following is an overview of the steps involved in reconciliation:

1. A SQL query or stored procedure is used to fetch target system records during reconciliation.
2. The scheduled job communicates to connector bundle and runs search operations over it, maps the task attributes to parameters of the reconciliation query or stored procedure, and then runs the query or stored procedure on the target system.
3. Target system records that meet the query or stored procedure criteria are fetched into Oracle Identity Manager.
4. If you have configured your target system as a trusted source, then each user record fetched from the target system is compared with existing OIM Users. The reconciliation rule is applied during the comparison process.

The next step of the process depends on the outcome of the matching operation:

- If a match is found between the target system record and the OIM User, then the OIM User attributes are updated with changes made to the target system record.
 - If no match is found, then the target system record is used to create an OIM User.
5. If you have configured your target system as a target resource, then each user record fetched from the target system is compared with existing target system resources assigned to OIM Users. The reconciliation rule is applied during the comparison process.

The next step of the process depends on the outcome of the matching operation:

- If a match is found between the target system record and a resource provisioned to an OIM User, then the database user resource is updated with changes made to the target system record.
- If no match is found, then the target system user record is compared with existing OIM Users. The next step depends on the outcome of the matching operation:

If a match is found, then the target system record is used to provision a resource for the OIM User.

If no match is found, then the status of the reconciliation event is set to No Match Found.

**Note:**

[Reconciliation Rules for MySQL](#) for information about the reconciliation rule

5.5.3 Target System Columns Used in Reconciliation from MySQL

As mentioned earlier in this guide, this connector can be configured to perform either target resource reconciliation or trusted source reconciliation. This section discusses the following topics:

- The `Lookup.DBUM.MySQL.UM.ReconAttrMap` lookup definition holds attribute mappings for user reconciliation. This lookup definition contains mapping of Oracle Identity Manager attributes and connector attributes.

See [Lookup.DBUM.MySQL.UM.ReconAttrMap](#) for more information.
- The `Lookup.DBUM.MySQL.UM.ReconAttrMap.Trusted` lookup definition holds attribute mappings for reconciliation in trusted mode. This lookup definition maps reconciliation profile attributes and connector attributes used in the reconciliation query. In addition, the connector attributes are associated to columns within the bundle.

See [Lookup.DBUM.MySQL.UM.ReconAttrMap.Trusted](#) for more information about this lookup definition.

5.5.4 Configuring the Target System As a Trusted Source

**Note:**

Skip this section if you do not want to designate the target system as a trusted source for reconciliation.

To configure trusted source reconciliation:

1. If you are using Oracle Identity Manager release 11.1.1.x:
 - a. Log in to the Administrative and User Console.
 - b. On the Welcome page, click **Advanced** in the upper-right corner of the page.
 - c. On the Welcome to Oracle Identity Manager Advanced Administration page, in the Configuration region, click **Create IT Resource**.

2. If you are using Oracle Identity Manager release 11.1.2.x or later:
 - a. Log in to Oracle Identity System Administration.
 - b. In the left pane, under Configuration, click **IT Resource**.
 - c. In the Manage IT Resource page, click **Create IT Resource**.
3. On the Step 1: Provide IT Resource Information page, enter the following information:
 - **IT Resource Name:** Enter a name for the IT resource. For example, `MySQL DB Trusted`.
 - **IT Resource Type:** Select the **MySQL DB** IT resource type for the IT resource.
4. Click **Continue**.
5. On the Step 2: Specify IT Resource Parameter Values page, specify values for the parameters of the IT resource.

Configuration Lookup: Name of the lookup definition in which you store the connector configuration information for the target system.

Sample Value: `Lookup.DBUM.MySQL.Configuration.Trusted`

Provide values for the other IT resource parameters.
6. Click **Continue**.

In the following steps, provide permissions on the IT resource that you are creating as per your requirements.

You can use this IT resource for trusted source reconciliation operations.

5.5.5 Reconciliation Rules for MySQL



See Also:

Reconciliation Metadata in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for generic information about reconciliation rules and reconciliation action rules

This section describes the reconciliation rules used by the reconciliation engine for this connector.

The following are the reconciliation rules for target resource reconciliation:

- **Rule name:** DBUM MySQL Target Recon
- **Rule element:** User Login Equals User Name

The following are the reconciliation rules for trusted source reconciliation:

- **Rule name:** MySQL DB Trusted
- **Rule element:** User Login Equal User ID

In these rule elements:

- User Login is the field on the OIM User form.

- User Name and User ID are the target system fields.

5.5.6 Viewing Reconciliation Rules for MySQL

After you deploy the connector, you can view the reconciliation rule for reconciliation by performing the following steps:



Note:

Perform the following procedure only after the connector is deployed.

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Development Tools**.
3. Double-click **Reconciliation Rules**.
4. Search for the rule name.

5.5.7 Reconciliation Action Rules for MySQL

Reconciliation action rules define that actions the connector must perform based on the reconciliation rules defined for Users.

[Table 5-14](#) lists the action rules for target resource reconciliation.

Table 5-14 Action Rules for Target Resource Reconciliation

Rule Condition	Action
No Matches Found	Assign to Administrator With Least Load
One Entity Match Found	Establish Link
One Process Match Found	Establish Link

[Table 5-15](#) lists the action rules for trusted source reconciliation.

Table 5-15 Action Rules for Trusted Source Reconciliation

Rule Condition	Action
No Matches Found	Create User
One Entity Match Found	Establish Link

5.5.8 Viewing Reconciliation Action Rules for MySQL

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

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Resource Management**.
3. Double-click **Resource Objects**.

4. Search for and open the resource object. The following are the names of the resource objects for each target system database:
 - Resource object for MySQL:
MySQL DB User
 - Resource object for MySQL as trusted source:
MySQL DB Trusted
5. Click the **Object Reconciliation** tab, and then click the **Reconciliation Action Rules** tab. The Reconciliation Action Rules tab displays the action rules defined for this connector.

5.5.9 Performing Full Reconciliation from MySQL

Full reconciliation involves reconciling all existing user records from the target system into Oracle Identity Manager. After you deploy the connector, you must first perform full reconciliation.

To perform a full reconciliation run, remove (delete) any value currently assigned to the Filter attribute and run one of the following scheduled jobs:

- For MySQL as a target resource: DBUM MySQL User Target Reconciliation
- For MySQL as a trusted source: DBUM MySQL User Trusted Reconciliation

See [Attributes for Scheduled Jobs for MySQL](#) for information about this scheduled job.

5.5.10 Performing Limited Reconciliation from MySQL

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

You can perform limited reconciliation by creating filters for the reconciliation module. This connector provides a Filter attribute (a scheduled task attribute) that allows you to use any of the DBUM resource attributes to filter the target system records. You can apply filters to the parent parameters in the reconciliation query file stored in a JAR file in the bundle directory of the connector installation media. For example, to locate the reconciliation query file, you can extract the `bundle/org.identityconnectors.dbum-1.0.1116.jar` file and open `scripts/mysql/Search.queries`.

The following table provides the description of the parent parameter that can be used with the Filter attribute of the scheduled jobs:

Parameter	Description
__UID__	Unique identity representing the user This parameter is mapped to USERNAME or __NAME__ connector attribute.

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

While deploying the connector, follow the instructions in [Configuring Scheduled Jobs for MySQL](#) to specify attribute values.

5.5.11 Performing Batched Reconciliation from MySQL

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

You can configure batched reconciliation to avoid these problems.

To configure batched reconciliation, you must specify value for the Batch Size reconciliation scheduled job attribute. Use this attribute to specify the number of records that must be included in each batch. By default, this value is empty.

If you specify a value other than `All`, then some of the newly added or modified user records may not get reconciled during the current reconciliation run. The following example illustrates this:

Suppose you specify the Batch Size value as `200` while configuring the scheduled jobs. Suppose that 314 user records were created or modified after the last reconciliation run. Of these 314 records, only 200 records would be reconciled during the current reconciliation run. The remaining 114 records would be reconciled during the next reconciliation run.

You specify values for the Batch Size attribute by following the instructions described in [Configuring Scheduled Jobs for MySQL](#).

5.6 Provisioning for MySQL

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

This section contains the following topics about provisioning:

- [Guidelines on Performing Provisioning Operations for MySQL](#)
- [Understanding the Provisioning Process for MySQL](#)
- [Configuring Direct Provisioning for MySQL](#)
- [Configuring Request-Based Provisioning for MySQL](#)
- [Enabling Request-Based Provisioning](#)
- [Switching Between Request-Based Provisioning and Direct Provisioning for MySQL](#)
- [Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x](#)

5.6.1 Guidelines on Performing Provisioning Operations for MySQL

The following are guidelines that you must apply while performing provisioning operations:

- Before you perform provisioning operations, lookup definitions must be synchronized with the lookup fields of the target system. In other words, run the scheduled jobs for lookup field synchronization before provisioning operations.
- Passwords for user accounts provisioned from Oracle Identity Manager must adhere to the password policy set in the target system.

- The character length of target system fields must be taken into account when specifying values for the corresponding Oracle Identity Manager fields.
- During an update password provisioning operation, ensure that you clear the existing text in the Password field, and then enter the new password.

5.6.2 Understanding the Provisioning Process for MySQL

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

When you install the connector on Oracle Identity Manager, the direct provisioning feature is automatically enabled. This means that the process form is enabled when you install the connector.

The following are types of provisioning operations:

- Direct provisioning
- Request-based provisioning
- Provisioning triggered by policy changes

If you configure the connector for request-based provisioning, then the process form is suppressed and the object form is displayed. In other words, direct provisioning is disabled when you configure the connector for request-based provisioning. If you want to revert to direct provisioning, then see [Switching Between Request-Based Provisioning and Direct Provisioning for MySQL](#).

The following is an overview of the Create User provisioning process in MySQL that is started through direct provisioning:

1. On the Create User page of the Administrative and User Console, the administrator enters the data required for an OIM User account creation.
Suppose the administrator enters the following values for the fields on the Create User page:
 - First Name: John
 - Last Name: Doe
 - User ID: jdoeAn OIM User account is created for John Doe.
2. The administrator selects the resource to be provisioned to the OIM User account that has been created. In this example, the administrator selects the MySQL DB User resource.
3. The administrator enters the data required for provisioning the MySQL DB User resource. Suppose the administrator wants to create a local user that requires a password to log in to the database. Therefore, the administrator enters the following values on the resource provisioning process form:
 - IT Resource: MySQL DB
 - User Name: JDoe
 - User Password: my_pa55word

In addition, the administrator also enters the following values on the process form for granting privileges:

- **Privilege:** `SELECT ON information_schema`
4. From the information available in the IT resource for the target system, the configuration (Lookup.DBUM.MySQL.Configuration) lookup definition is identified. This lookup definition stores configuration information that is used during connector operations.
 5. The connector bundle contains the script (Provisioning.queries) required for provisioning operations.
 6. The identifiers in the SQL statement are replaced with the input parameters fetched from the query. Then, the SQL statement with actual values is formed.
 7. The connector runs the SQL statement on MySQL and creates the jdoe account on the target system. The next step of the process depends on whether the administrator had entered data for granting privileges to the target system account.

If the administrator did not enter any values for granting privileges, then the provisioning process ends here. Otherwise, the process continues to the next step.

8. While performing Step 3, the administrator had entered the required data for granting privileges to the jdoe account. Therefore, the corresponding query as mentioned in Step 6 is read.
9. The complete SQL statement that must be run to perform the Add privilege provisioning operation is formed.
10. The input parameters required to run the SQL statement are fetched from the parameter configuration done using the queries in the query files.
11. The identifiers in the SQL statement (formed in Step 9) are replaced with the input parameters fetched from the query. Then, the SQL statement with actual values is formed.
12. The query runs the SQL statement on the target system (MySQL) and grants privileges to the jdoe target system account.

5.6.3 Configuring Direct Provisioning for MySQL

In direct provisioning, the Oracle Identity Manager administrator uses the Administrative and User Console to create a target system account for a user.

To provision a resource by using the direct provisioning approach:

1. Log in to the Administrative and User Console.
2. To first create an OIM User before provisioning a database account to the user:
 - a. On the Welcome to Identity Administration page, in the Users region, click **Create User**.
 - b. On the Create User page, enter values for the OIM User fields, and then click the save icon.
3. To search for an existing OIM User to be provisioned:
 - a. On the Welcome to Identity Administration page, search for the user by selecting **Users** from the Search list on the left pane.

Alternatively, in the Users region, click **Advanced Search - User**, provide a search criterion, and then click **Search**.

- Direct provisioning cannot be used if you enable request-based provisioning.

5.6.4.2 Enabling Request-Based Provisioning

The following sections provide information about the procedures you must perform to enable request-based provisioning:



Note:

The procedure described in this section is applicable only if you are using Oracle Identity Manager release 11.1.1.x.

- [Approver's Role in Request-Based Provisioning for MySQL](#)
- [Importing MySQL Request Datasets Using Deployment Manager](#)
- [End User's Role in Request-Based Provisioning for MySQL](#)
- [Enabling the Auto Save Form Feature for MySQL](#)
- [Running the PurgeCache Utility for MySQL](#)

5.6.4.2.1 Approver's Role in Request-Based Provisioning for MySQL

The following are steps performed by the approver in a request-based provisioning operation:

1. Log in to the Administrative and User Console.
2. On the Welcome page, click **Self-Service** in the upper-right corner of the page.
3. On the Welcome to Identity Manager Self Service page, click the **Tasks** tab.
4. On the **Approvals** tab, in the first section, you can specify a search criterion for request task that is assigned to you.
5. From the search results table, select the row containing the request you want to approve, and then click **Approve Task**.

A message confirming that the task was approved is displayed.

5.6.4.2.2 Importing MySQL Request Datasets Using Deployment Manager

A request dataset is an XML file that specifies the information to be submitted by the requester during a provisioning operation. These request datasets specify information about the default set of attributes for which the requester must submit information during a request-based provisioning operation.

To import a request dataset XML file by using the Deployment Manager:

1. Log in to the Oracle Identity Manager Administrative and User Console.
2. Click the **Deployment Management** link on the left navigation bar.
3. Click the **Import** link under Deployment Management.

A dialog box for opening files is displayed.

4. Locate and open the request dataset XML file, `DBUserManagement-MySQL-Datasets.xml`, which is in the `xml` directory of the installation media.

Details of this XML file are shown on the **File Preview** page.

5. Click **Add File**.

The Substitutions page is displayed.

6. Click **Next**.

The Confirmation page is displayed.

7. Click **Import**.

8. Close the Deployment Manager dialog box.

The request dataset is imported into Oracle Identity Manager.

5.6.4.2.3 End User's Role in Request-Based Provisioning for MySQL

The following steps are performed by the end user in a request-based provisioning operation:

1. Log in to the Administrative and User Console.
2. On the Welcome page, click **Advanced** in the upper-right corner of the page.
3. On the Welcome to Identity Administration page, click the **Administration** tab, and then click the **Requests** tab.

4. From the Actions menu on the left pane, select **Create Request**.

The Select Request Template page is displayed.

5. From the Request Template list, select **Provision Resource** and click **Next**.

6. On the Select Users page, specify a search criterion in the fields to search for the user that you want to provision the resource, and then click **Search**. A list of users that match the search criterion you specify is displayed in the Available Users list.

7. From the **Available Users** list, select the user to whom you want to provision the account.

If you want to create a provisioning request for more than one user, then from the **Available Users** list, select users to whom you want to provision the account.

8. Click **Move** or **Move All** to include your selection in the Selected Users list, and then click **Next**.

9. On the Select Resources page, click the arrow button next to the Resource Name field to display the list of all available resources.

10. From the Available Resources list, select **MySQL DB User**, move it to the Selected Resources list, and then click **Next**.

11. On the Resource Details page, enter details of the account that must be created on the target system, and then click **Next**.

12. On the Justification page, you can specify values for the following fields, and then click **Finish**.

- Effective Date
- Justification

A message confirming that your request has been sent successfully is displayed along with the Request ID.

13. If you click the request ID, then the Request Details page is displayed.

14. To view details of the approval, on the Request Details page, click the **Request History** tab.

5.6.4.2.4 Enabling the Auto Save Form Feature for MySQL

To enable the Auto Save Form feature:

1. Log in to the Design Console.
2. Expand **Process Management**, and then double-click **Process Definition**.
3. Search for and open the **MySQL DB** process definition.
4. Select the **Auto Save Form** check box.
5. Click the save icon.

5.6.4.2.5 Running the PurgeCache Utility for MySQL

Run the PurgeCache utility to clear content belonging to the Metadata category from the server cache.

See [Clearing Content Related to the Connector Resource Bundles from the Server Cache](#) for instructions.

The procedure to enable enabling request-based provisioning ends with this step.

5.6.5 Switching Between Request-Based Provisioning and Direct Provisioning for MySQL

If you have configured the connector for request-based provisioning, you can always switch to direct provisioning. Similarly, you can always switch back to request-based provisioning any time. This section discusses the following topics:

- [Switching From Request-Based Provisioning to Direct Provisioning](#)
- [Switching From Direct Provisioning to Request-Based Provisioning](#)

5.6.5.1 Switching From Request-Based Provisioning to Direct Provisioning



Note:

It is assumed that you have performed the procedure described in [Configuring Request-Based Provisioning for MySQL](#).

If you want to switch from request-based provisioning to direct provisioning, then:

1. Log in to the Design Console.
2. Disable the Auto Save Form feature as follows:
 - a. Expand **Process Management**, and then double-click **Process Definition**.
 - b. Search for and open the **MySQL DB** process definition.
 - c. Deselect the Auto Save Form check box.

- d. Click the save icon.
3. If the Self Request Allowed feature is enabled, then:
 - a. Expand **Resource Management**, and then double-click **Resource Objects**.
 - b. Search for and open the **MySQL DB User** resource object.
 - c. Deselect the Self Request Allowed check box.
 - d. Click the save icon.

5.6.5.2 Switching From Direct Provisioning to Request-Based Provisioning

If you want to switch from direct provisioning back to request-based provisioning, then:

1. Log in to the Design Console.
2. Enable the Auto Save Form feature as follows:
 - a. Expand **Process Management**, and then double-click **Process Definition**.
 - b. Search for and open the **MySQL DB** process definition.
 - c. Select the **Auto Save Form** check box.
 - d. Click the save icon.
3. If you want to enable end users to raise requests for themselves, then:
 - a. Expand **Resource Management**, and then double-click **Resource Objects**.
 - b. Search for and open the **MySQL DB User** resource object.
 - c. Select the Self Request Allowed check box.
 - d. Click the save icon.

5.6.6 Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x

To perform provisioning operations in Oracle Identity Manager release 11.1.2.x:

1. Log in to Identity Self Service.
2. If you want to first create an OIM User and then provision a target system account, then:

 **Note:**

See *Creating Users in Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Manager* for more information about creating a user.

- a. In the left pane, under Administration, click **Users**.
The Search Users page is displayed.
- b. From the Actions menu, select **Create**. Alternatively, you can click **Create** on the toolbar.

- c. On the Create User page, enter values for the OIM User fields, and then click **Submit**. A message is displayed stating that the user is created successfully.
3. If you want to provision a target system account to an existing OIM User, then:

 **Note:**

See Searching Users in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Manager* for more information about searching a user.

- a. In the left pane, under Administration, click **Users**.
The Search Users page is displayed.
 - b. Specify a search criteria to search for the OIM User, and then click **Search**.
 - c. From the list of users displayed in the search results, select the OIM User. The user details page is displayed on the right pane.
4. On the Account tab, click **Request Accounts**.
5. In the Catalog page, search for and add to cart the application instance (in other words, the account to be provisioned), and then click **Checkout**.
6. Specify value for fields in the application form and then click **Ready to Submit**.
7. Click **Submit**.
8. If you want to provision entitlements, then:
 - a. On the Entitlements tab, click **Request Entitlements**.
 - b. In the Catalog page, search for and add to cart the entitlement, and then click **Checkout**.
 - c. Click **Submit**.

5.7 Extending the Connector for MySQL

The following sections describe procedures that you can perform to extend the functionality of the connector for addressing your specific business requirements:

 **Note:**

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

- [Modifying Predefined Queries or Creating New Queries for MySQL](#)
- [Configuring Queries to Add Support for Custom Parameters and Lookup Fields for MySQL](#)
- [About Configuring the Connector for Multiple Installations of MySQL](#)

- [About Configuring the Connector for Multiple Trusted Source Reconciliation from MySQL](#)
- [Configuring Validation of Data During Reconciliation and Provisioning for MySQL](#)
- [Configuring Transformation of Data During User Reconciliation for MySQL](#)
- [Configuring Resource Exclusion Lists for MySQL](#)
- [Setting Up Action Scripts for MySQL](#)

5.7.1 Modifying Predefined Queries or Creating New Queries for MySQL

The following sections discuss the syntax and guidelines that you must apply while modifying the predefined queries or creating new queries:

- [About the Queries for MySQL Database](#)
- [Syntax of Provisioning Queries for MySQL Database](#)
- [Syntax of Reconciliation Queries for MySQL Database](#)
- [Syntax of List of Values Queries for MySQL Database](#)

5.7.1.1 About the Queries for MySQL Database

Predefined queries are provided to reconcile target system user records, synchronize lookup field values with Oracle Identity Manager, and for provisioning operations. You can modify the predefined queries or add your own queries.

The query files are included in a JAR file in the bundle directory of the connector installation media. For example, `bundle/org.identityconnectors.dbum-1.0.1116.jar`.

The connector includes the following types of queries:

- **Provisioning Queries**
They are used for create, update, and delete operations. The query file is `scripts/mysql/Provisioning.queries`.
- **List of Values Search Queries**
They are used for reconciliation of lookup definitions. A list of value query operates on a set of values for fields such as profiles, privileges, roles, and tablespaces. The query file is `scripts/mysql/LoVSearch.queries`.
- **Account Search Queries**
They are used for full and delete reconciliation operations. An account search query operates on account and group searches with various conditions. The query file is `scripts/mysql/Search.queries`.

 **Note:**

The stored procedure OUT parameters cannot be configured for write-back on the process form. The returned values cannot be used for any connector operations.

5.7.1.2 Syntax of Provisioning Queries for MySQL Database

The following is the syntax of the queries used for provisioning operations:

```

QUERYID {
  Query="QUERY"
  QueryType="QUERYTYPE"
  Parameters=["PARAM1":"PARAMDEFN1", "PARAM2":"PARAMDEFN2" ...]
  ExtensionJoin="EXTENSIONJOIN"
  ExtensionSeparator="EXTENSIONSEPARATOR"
  QueryExtensions=["EXTENSION1", "EXTENSION2" ...]
}

```

For example:

```

CREATE_USER {
  Query="CREATE USER {__NAME__} IDENTIFIED BY {__PASSWORD__}"
  QueryType="SQL"

  Parameters=["__NAME__": "Type:String", "__PASSWORD__": "Type:GuardedString, TAGS:QUOTES"]
  QueryExtensions=[]
}

```

In this syntax:

- *QUERYID* refers to the unique name of the query.
For example: `CREATE_USER`
- *QUERY* refers to the main query.
For example: `Query="CREATE USER {__NAME__} IDENTIFIED BY {__PASSWORD__}"`
- *QueryType* refers to the type of the main query, either an SQL query or a stored procedure. The value of *QUERYTYPE* can be `SQL` or `StoredProc`.
For example: `QueryType="SQL"`
- *Parameters* refers to the list of comma separated parameters and parameter definitions used with the main query, represented by `"PARAM1":"PARAMDEFN1"`, `"PARAM2":"PARAMDEFN2"`, and so on.

For example:

```

Parameters=["__NAME__": "Type:String", "__PASSWORD__": "Type:GuardedString, TAGS:QUOTES"]

```

A parameter can have the following attributes:

- *Type* is the type of the parameter.

- Direction is the flow of data from the query to or from the parameter. It can have a value of `IN`, `OUT`, or `INOUT`.
- TAGS is the enclosure characters that are applied to each parameter before the query is processed. It can have a value of `DOUBLEQUOTES`, `QUOTES`, `UPPERCASE`, or `LOWERCASE`.

If you want to use multiple tags, you must encapsulate the tags in escaped quotes and separate them by commas. However, you must not use `DOUBLEQUOTES` with `QUOTES` or `UPPERCASE` with `LOWERCASE` in the same query.

For example: `"Type:String,TAGS:\"DOUBLEQUOTES,UPPERCASE\""`

- ExtensionJoin (optional) refers to the operator, represented by `EXTENSIONJOIN`, used to join the main query with query extensions.

For example: `ExtensionJoin=","`

- ExtensionSeparator (optional) refers to the delimiter between query extensions, represented by `EXTENSIONSEPARATOR`.

For example: `ExtensionSeparator=","`

- QueryExtensions (optional) refers to the extensions that must be appended to the main query, represented by `EXTENSION1`, `EXTENSION2`, and so on.

During a provisioning operation, the connector combines all these components to the following query:

```
QUERY PARAM1, PARAM2... [EXTENSIONJOIN [EXTENSION1
EXTENSIONSEPARATOR EXTENSION2 EXTENSIONSEPARATOR...]]
```

For example:

```
CREATE USER {__NAME__} IDENTIFIED BY {__PASSWORD__}
```

Table 5-16 lists the script selection logic of the provisioning queries:

Table 5-16 Script Section Logic for MySQL Provisioning Queries

Operation	Selection Logic	Query IDs
CREATE	CREATE_OBJECTTYPE	CREATE_USER
DELETE	DELETE_OBJECTTYPE	DELETE_USER
RESET PASSWORD	SET_PASSWORD	SET_PASSWORD
ADD CHILD VALUES	UPDATE_ADD_ATTRIBUTE	UPDATE_ADD_PRIVILEGES
REMOVE CHILD VALUES	UPDATE_REVOKE_ATTRIBUTE	UPDATE_REVOKE_PRIVILEGES

5.7.1.3 Syntax of Reconciliation Queries for MySQL Database

The following is the syntax of the search queries used during reconciliation operations:

```
QUERYID {
Query="QUERY"
QueryType="QUERYTYPE"
```

```
Parameters=["PARAM1":"PARAMDEFN1", "PARAM2":"PARAMDEFN2"...]
ExtensionJoin="EXTENSIONJOIN"
ExtensionSeparator="EXTENSIONSEPARATOR"
QueryExtensions=["EXTENSION1","EXTENSION2"...]
}
```

For example:

```
SEARCH_USER {
  Query="SELECT {__UID__} FROM MYSQL.USER {filter}"
  QueryType="SQL"
  Parameters=["__UID__": "Type:String,Direction:OUT,ColName:USER"]
  QueryExtensions=["SEARCH_USER_PRIVILEGE"]
}
```

In this syntax:

- *QUERYID* refers to the unique name of the query.

For example: `SEARCH_USER`

QUERYID can be one of the following values:

- `SEARCH_USER`
- `BATCHED_SEARCH_USER`
- `SEARCH_USER_PRIVILEGE`

- *QUERY* refers to the main query.

For example: `Query="SELECT {__UID__} FROM MYSQL.USER {filter}"`

- *QueryType* refers to the type of the main query, either an SQL query, a stored procedure, or a query extension. The value of *QUERYTYPE* can be `SQL`, `StoredProc`, or `QUERYEXTENSION`.

For example: `QueryType="SQL"`

- *Parameters* refers to the list of comma separated parameters and parameter definitions used with the main query, represented by `"PARAM1":"PARAMDEFN1"`, `"PARAM2":"PARAMDEFN2"`, and so on.

For example:

```
Parameters=["__UID__": "Type:String,Direction:OUT,ColName:USER"]
```

A parameter can have the following attributes:

- *Type* is the type of the parameter.
- *Direction* is the flow of data from the query to or from the parameter. It can have a value of `IN`, `OUT`, or `INOUT`.
- *ColName* is the column name in the target system corresponding to the parameter in the query.
- *ColQuery* is the query used to fetch values for the corresponding query parameter.
- *ExtensionJoin* (optional) refers to the operator, represented by `EXTENSIONJOIN`, used to join the main query with query extensions.

For example: `ExtensionJoin=", "`

- **ExtensionSeparator** (optional) refers to the delimiter between query extensions, represented by *EXTENSIONSEPARATOR*.
For example: `ExtensionSeparator=", "`
- **QueryExtensions** (optional) refers to the extensions that must be appended to the main query, represented by *EXTENSION1*, *EXTENSION2*, and so on.
For example: `QueryExtensions=["SEARCH_USER_PRIVILEGE"]`

During a reconciliation operation, the connector combines all these components to the following query:

```
QUERY PARAM1, PARAM2... [EXTENSIONJOIN [EXTENSION1
EXTENSIONSEPARATOR EXTENSION2 EXTENSIONSEPARATOR...]]
```

For example:

```
SELECT {__UID__} FROM MYSQL.USER {filter} SEARCH_USER_PRIVILEGE
```

5.7.1.4 Syntax of List of Values Queries for MySQL Database

If a search query is performed on account types, such as User Name, then the query is considered as a reconciliation query. If a search query is performed on any other object, then the query is considered as a list of values query.

The following is the syntax of the list of values queries used for lookup field synchronization:

OBJECTTYPE = "QUERY"

For example:

```
__PRIVILEGES__="SELECT CONCAT(p.PRIVILEGE_TYPE, ' ON ',s.SCHEMA_NAME)
SchemaPrivilege FROM INFORMATION_SCHEMA.SCHEMATA
s,INFORMATION_SCHEMA.SCHEMA_PRIVILEGES p"
```

In this syntax:

- **OBJECTTYPE** refers to the lookup field attribute.
For example: `__PRIVILEGES__`
- **QUERY** refers to the query used for fetching a lookup field attribute.
For example: `SELECT CONCAT(p.PRIVILEGE_TYPE, ' ON ',s.SCHEMA_NAME)
SchemaPrivilege FROM INFORMATION_SCHEMA.SCHEMATA
s,INFORMATION_SCHEMA.SCHEMA_PRIVILEGES p`

The list of values queries return values that are used as lookup field entries. By default, the connector includes dedicated scheduled job for each lookup definition. To use a custom lookup definition, you must add custom fields in the query file.

5.7.2 Configuring Queries to Add Support for Custom Parameters and Lookup Fields for MySQL

The connector uses preconfigured queries for connector operations such as create, delete, and search. You can add custom parameters and lookup definition fields as per your requirements.

The procedure to add a parameter or a lookup definition field to a query file is discussed in the following sections:

- [Updating the Query Files for MySQL Database](#)
- [Configuring Oracle Identity Manager](#)

5.7.2.1 Updating the Query Files for MySQL Database

To update the query files:

1. If the connector is already installed, run the Oracle Identity Manager Download JARs utility to download the connector bundle JAR file from the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

```
OIM_HOME/server/bin/DownloadJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/DownloadJars.sh
```

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

2. Copy the bundle JAR file in a temporary directory.

Sample JAR file: `bundle/org.identityconnectors.dbum-1.0.1116.jar`

Sample temporary directory: `c:\temp`

3. Run the following command to extract the connector bundle JAR file:

```
jar -xvf org.identityconnectors.dbum-1.0.1116.jar
```

 **Note:**

You can also run the WinZip or WinRAR utility to extract the contents from the JAR file.

4. Delete the bundle JAR file in the temporary directory.
5. Update the value of **ConnectorBundle-Version** in the manifest file, `META-INF/MANIFEST.MF`, to a new value.

For example:

```
ConnectorBundle-Version: 1.0.1117
```

6. Depending on your requirement, update the query files with new parameters as per the query syntax described in [Modifying Predefined Queries or Creating New Queries for MySQL](#).

For example, if you want to add a new parameter, `CUSTOM_ATTRIBUTE`, to the `CREATE_USER` provisioning query:

- a. Open the provisioning query file in a text editor.

Sample query file:

```
c:\temp\bundle\org.identityconnectors.dbum-1.0.1116\scripts\mysql\Provisioning.queries
```

- b. Add the parameter, `CUSTOM_ATTRIBUTE`, to the `CREATE_USER` query.

The following is a sample updated query:

```
CREATE_USER {
  Query="CREATE USER {__NAME__} IDENTIFIED BY {__PASSWORD__},
{CUSTOM_ATTRIBUTE}"
  QueryType="SQL"
  Parameters=["__NAME__": "Type:String",
  "__PASSWORD__": "Type:GuardedString,TAGS:QUOTES",
  "CUSTOM_ATTRIBUTE": "Type:String,Direction:IN"]
  QueryExtensions=[]
}
```

- c. Save and close the query file.
7. Create a new bundle JAR file that contains the updated manifest file and the provisioning query file as follows:

- a. Open the command prompt and navigate to the temporary directory:

```
c:\temp
```

- b. Run the following command:

```
jar -cvfm org.identityconnectors.dbum-1.0.1117.jar *
```

The new connector bundle JAR name contains the new bundle version.

8. In the case of a remote connector server, copy the new bundle JAR file in the bundles directory of the remote connector server, instead of posting the JAR file to the Oracle Identity Manager database. Skip to Step 10.
9. Run the Oracle Identity Manager Update JARs utility to update the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

If you have installed both the Oracle and MySQL connectors on the same Oracle Identity Manager, then ensure that all third-party JAR files are part of the `/lib` directory in the connector bundle JAR file.

For Microsoft Windows:

```
OIM_HOME/server/bin/UpdateJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UpdateJars.sh
```

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

10. Update the configuration lookup with the new bundle version.

For example, you can update the Lookup.DBUM.MySQL.Configuration lookup definition.

5.7.2.2 Configuring Oracle Identity Manager

You can skip this procedure if the parameter you added already exists as a default form field in Oracle Identity Manager.

To configure Oracle Identity Manager for adding a parameter:

1. Log into Oracle Identity Manager Design Console.
2. Create a new version of the process form:
 - a. Expand **Development Tools**.
 - b. Double-click **Form Designer**.
 - c. Search for and open the **UD_DB_MYS_U** process form.
 - d. Click **Create New Version**.

On the Create a new version dialog box, enter a new version in the Label field, and then click the save icon.

3. Add the new field on the process form.
 - a. Click **Add**.

A field is added to the list. Enter the details of the field.

For example, if you are adding the CustomAttribute1 field, enter `UD_DB_MYS_U_CUSTOM1` in the **Name** field and then enter the rest of the details of this field.
 - b. Click the save icon and then click **Make Version Active**.
4. If you are using Oracle Identity Manager release 11.1.2.x or later, then all changes made to the Form Designer of the Design Console must be done in a new UI form as follows:
 - a. Log in to Oracle Identity System Administration.
 - b. Create and active a sandbox.
 - c. Create a new UI form to view the newly added field along with the rest of the fields. See *Creating Forms By Using the Form Designer in Oracle Fusion Middleware Administering Oracle Identity Manager*.
 - d. Associate the newly created UI form with the application instance of your target system. To do so, open the existing application instance for your resource, from the Form field, select the form (created in Step 4.c), and then save the application instance.
 - e. Publish the sandbox as described in *Publishing a Sandbox of Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

5. Create an entry for the field in the lookup definition for provisioning as follows:
 - a. Expand **Administration**.
 - b. Double-click **Lookup Definition**.
 - c. Search for and open the **Lookup.DBUM.MySQL.UM.ProvAttrMap** lookup definition.
 - d. Click **Add** and enter the Code Key and Decode values for the field.
 The Code Key value must be the form field name. The Decode value must be the attribute name on the target system.
 For example, enter `Custom Attribute 1` in the **Code Key** field and then enter `CustomAttribute1` in the **Decode** field.
 - e. Click the save icon.
6. Create a process task to update the new field Custom Attribute 1 as follows:
 - a. Expand **Process Management**.
 - b. Double-click **Process Definition** and open the **MySQL DB User** process definition.
 - c. Click **Add** and enter the task name, for example, `Custom Attribute 1 Updated`, and the task description.
 - d. In the process definition, select **Conditional** and **Allow Multiple Instances** fields and click the save icon.
 - e. On the **Integration** tab, click **Add**, and then click **Adapter**.
 - f. Select the **adpMYSQLDBUMUPDATEUSER** adapter, click the save icon, and then click **OK** in the message that is displayed.
 - g. To map the adapter variables listed in this table, select the adapter, click **Map**, and then specify the data given in the following table:

Variable Name	Data Type	Map To	Qualifier	Literal Value
Adapter return value	Object	Response code	NA	NA
attributeName	String	Literal	String	Custom Attribute 1
itRes	String	Literal	String	UD_DB_MYS_U_ITRES
objectType	String	Literal	String	User
processInstanceKey	Long	Process Data	Process Instance	NA

- h. On the **Responses** tab, click **Add** to add the following response codes:

Code Name	Description	Status
ERROR	Error occurred	R
UNKNOWN	An unknown response was received	R
SUCCESS	Operation completed	C

- i. Click the save icon and then close the dialog box.

5.7.3 About Configuring the Connector for Multiple Installations of MySQL

You might want to configure the connector for multiple installations of the target system. The following example illustrates this requirement:

The London and New York offices of Example Multinational Inc. have their own installations of the target system. The company has recently installed Oracle Identity Manager, and they want to configure Oracle Identity Manager to link all the installations of the target system.

To meet the requirement posed by such a scenario, you can create copies of connector objects, such as the IT resource and resource object.

The decision to create a copy of a connector object might be based on a requirement. For example, an IT resource can hold connection information for one target system installation. Therefore, it is mandatory to create a copy of the IT resource for each target system installation.

With some other connector objects, you do not need to create copies at all. For example, a single attribute-mapping lookup definition can be used for all installations of the target system.

All connector objects are linked. For example, a scheduled job holds the name of the IT resource. Similarly, the IT resource for a target system such as MySQL holds the name of the configuration lookup definition, `Lookup.DBUM.MySQL.Configuration`. If you create a copy of an object, then you must specify the name of the copy in associated connector objects.

Note:

- To reconcile data from a particular target system installation, specify the name of the IT resource for that target system installation as the value of the scheduled job attribute that holds the IT resource name. For example, you enter the name of the IT resource as the value of the IT resource attribute of the scheduled job that you run.
- When you use Identity Self Service to perform provisioning, you can specify the IT resource corresponding to the target system installation to which you want to provision the user.

[Table 5-17](#) lists associations between connector objects whose copies can be created and the other objects that reference these objects. When you create a copy of a connector object, use this information to change the associations of that object with other objects.

 **Note:**

- On a particular Oracle Identity Manager installation, if you create a copy of a connector object, then you must set a unique name for it.
- If you are using Oracle Identity Manager release 11.1.2.x or later, then in addition to the procedure described in this section, you must create an application instance for each IT resource. See [Configuring Oracle Identity Manager Release 11.1.2 or Later](#) for information on creating an application instance.

Table 5-17 Connector Objects and Their Associations

Connector Object	Name	Referenced By	Comments on Creating a Copy
IT resource	MySQL DB	<ul style="list-style-type: none"> • UD_DB_MYS_U (process form) • Scheduled tasks 	Create a copy of the IT resource with a different name.
Resource object	MySQL DB User MySQL DB Trusted	All connector operations	<p>It is optional to create a copy of the resource object. If you are reconciling the same set of attributes from all installations of the target system, then you need not create a copy of the resource object.</p> <p>Note: Create copies of the resource object only if there are differences in attributes between the various installations of the target system.</p>
Scheduled Jobs	There are many scheduled jobs for different purposes.	NA	You can use the scheduled jobs with the same names. However, you must update the values of the parameters depending on the target system you want to use.
Process definition	MySQL DB User	NA	<p>It is optional to create a copy of the process definition. If you are reconciling or provisioning the same set of attributes from all installations of the target system, then you need not create a copy of the process definition.</p> <p>Note: Create copies of the process form only if there are differences in attributes between the various installations of the target system.</p>
Process form	UD_DB_MYS_U	MySQL DB User (Process definition)	<p>It is optional to create a copy of the process form. If you are provisioning the same set of attributes from all installations of the target system, then you need not create a copy of the process definition.</p> <p>Note: Create copies of the process form only if there are differences in attributes between the various installations of the target system.</p>

Table 5-17 (Cont.) Connector Objects and Their Associations

Connector Object	Name	Referenced By	Comments on Creating a Copy
Child process form	<ul style="list-style-type: none"> UD_DB_MYS_P 	<ul style="list-style-type: none"> MySQL DB User (Process definition) UD_DB_MYS_U (Process form) 	It is optional to create a copy of the child process form. If you are provisioning a new set of child data, then you need to create a copy of the child and parent process forms. Then, assign the newly created child process form to the newly created parent process form.
Configuration lookup definition for a target system configured as a target resource	Lookup.DBUM.MySQL.Configuration	MySQL DB (IT resource)	<p>It is optional to create a copy of the configuration lookup definition. If you are provisioning and reconciling the same set of attributes in all installations of the target system (configured as a target resource), then you need not create a copy of the configuration lookup definition.</p> <p>Note: Create copies of the configuration lookup definition only if there are differences in attributes between the various installations of the target system and you have created a new process form.</p>
Configuration lookup definition for a target system configured as a trusted source	Lookup.DBUM.MySQL.Configuration.Trusted	MySQL DB (IT resource)	<p>It is optional to create a copy of the configuration lookup definition. If you are reconciling the same set of attributes in all installations of the target system (configured as a trusted source), then you need not create a copy of the configuration lookup definition.</p> <p>Note: Create copies of the configuration lookup definition for trusted source only if there are differences in attributes between the various installations of the target system and you have created a new process form.</p>
Resource object attributes mapping lookup definition (for target resource)	Lookup.DBUM.MySQL.UM.ReconAttrMap	NA	<p>It is optional to create a copy of resource object attribute mapping lookup definition. If you are reconciling the same set of attributes in all installations of the target system, then you need not to create a copy of resource object attribute mapping lookup.</p> <p>Note: Create copies of this lookup definition only if there are differences in attributes between the two installations of the target system.</p>

Table 5-17 (Cont.) Connector Objects and Their Associations

Connector Object	Name	Referenced By	Comments on Creating a Copy
Configuration lookup definition for a target system configured as a trusted source	Lookup.DBUM.MySQL.UM.ReconAttrMap.Trusted	MySQL DB (IT resource)	<p>It is optional to create a copy of the configuration lookup definition. If you are reconciling the same set of attributes in all installations of the target system (configured as a trusted source), then you need not create a copy of the configuration lookup definition.</p> <p>Note: Create copies of the configuration lookup definition for trusted source only if there are differences in attributes between the various installations of the target system and you have created a new process form.</p>

5.7.4 About Configuring the Connector for Multiple Trusted Source Reconciliation from MySQL

Note:

This connector supports multiple trusted source reconciliation.

This section describes an optional procedure. Perform this procedure only if you want to configure the connector for multiple trusted source reconciliation.

The following are examples of scenarios in which there is more than one trusted source for user data in an organization:

- One of the target systems is a trusted source for data about users. The second target system is a trusted source for data about contractors. The third target system is a trusted source for data about interns.
- One target system holds the data of some of the identity fields that constitute an OIM User. Two other systems hold data for the remaining identity fields. In other words, to create an OIM User, data from all three systems would need to be reconciled.

If the operating environment of your organization is similar to that described in either one of these scenarios, then this connector enables you to use the target system as one of the trusted sources of person data in your organization.

See *Managing Reconciliation in Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed information about multiple trusted source reconciliation.

5.7.5 Configuring Validation of Data During Reconciliation and Provisioning for MySQL

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

To configure validation of data:

1. Write code that implements the required validation logic in a Java class with a fully qualified domain name (FQDN), such as `org.identityconnectors.dbum.extension.DBUMValidator`.

This validation class must implement the `validate` method. The following sample validation class checks if the value in the First Name attribute contains the number sign (#):

```
package com.validationexample;

import java.util.HashMap;

public class MyValidator {
    public boolean validate(HashMap hmUserDetails, HashMap hmEntitlementDetails,
        String sField) throws ConnectorException {

        /* You must write code to validate attributes. Parent
         * data values can be fetched by using hmUserDetails.get(field)
         * For child data values, loop through the
         * ArrayList/Vector fetched by hmEntitlementDetails.get("Child
Table")

         * Depending on the outcome of the validation operation,
         * the code must return true or false.
         */

        /*
         * In this sample code, the value "false" is returned if the field
         * contains the number sign (#). Otherwise, the value "true" is
         * returned.
         */
        boolean valid = true;
        String sFirstName = (String) hmUserDetails.get(sField);
        for (int i = 0; i < sFirstName.length(); i++) {
            if (sFirstName.charAt(i) == '#') {
                valid = false;
                break;
            }
        }
        return valid;
    }
}
```

2. Log in to the Design Console.
3. Search for and open one of the lookup definitions (or create a new lookup) listed in [Lookup Definition for Validation of Data in MySQL](#).

For example, **Lookup.DBUM.MySQL.UM.ProvValidations**.

 **Note:**

If you cannot find these lookup definitions, create new lookup definitions.

4. In the **Code Key** column, enter the resource object field name that you want to validate. For example, `Username`.
5. In the **Decode** column, enter the class name. For example, `org.identityconnectors.dbum.extension.DBUMValidator`.
6. Save the changes to the lookup definition.
7. Search for and open the configuration lookup definition for the target system you use.

For example, **Lookup.DBUM.MySQL.UM.Configuration**.

8. In the **Code Key** column, enter one of the following entries:
 - To configure validation of data for reconciliation:
`Recon Validation Lookup`
 - To configure validation of data for provisioning:
`Provisioning Validation Lookup`
9. In the **Decode** column, enter the name of the lookup you updated or created in step 3.
For example, **Lookup.DBUM.MySQL.UM.ProvValidations**.
10. Save the changes to the lookup definition.
11. Create a JAR with the class and upload it to the Oracle Identity Manager database as follows:

Run the Oracle Identity Manager Upload JARs utility to post the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

```
OIM_HOME/server/bin/UploadJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UploadJars.sh
```

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

12. Run the PurgeCache utility to clear content related to request datasets from the server cache.
13. Perform reconciliation or provisioning to verify validation for the field, for example, Username.

5.7.6 Configuring Transformation of Data During User Reconciliation for MySQL

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

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

1. Write code that implements the required transformation logic in a Java class with a fully qualified domain name (FQDN), such as `org.identityconnectors.dbum.extension.DBUMTransformation`.

This transformation class must implement the transform method. The following sample transformation class modifies the Username attribute by using values fetched from the `__NAME__` attribute of the target system:

```
package com.transformationexample;

import java.util.HashMap;

public class MyTransformer {
    public Object transform(HashMap hmUserDetails, HashMap hmEntitlementDetails,
String sField) throws ConnectorException {
        /*
         * You must write code to transform the attributes.
         * Parent data attribute values can be fetched by
         * using hmUserDetails.get("Field Name").
         * To fetch child data values, loop through the
         * ArrayList/Vector fetched by hmEntitlementDetails.get("Child
Table")
         * Return the transformed attribute.
         */
        String sUserName = (String) hmUserDetails.get("__NAME__");
        return sUserName + "@example.com";
    }
}
```

2. Log in to the Design Console.
3. Search for and open one of the lookup definitions (or create a new lookup) listed in [Lookup Definitions for Transformation of Data in MySQL](#).

For example, **Lookup.DBUM.MySQL.UM.ReconTransformations**.

Note:

If you cannot find these lookup definitions, create new lookup definitions.

4. In the **Code Key** column, enter the resource object field name you want to transform. For example, `Username`.
5. In the **Decode** column, enter the class name. For example, `org.identityconnectors.dbum.extension.DBUMTransformation`.
6. Save the changes to the lookup definition.
7. Search for and open the **Lookup.DBUM.MySQL.UM.Configuration** lookup definition.
8. In the **Code Key** column, enter `Recon Transformation Lookup`.
9. In the **Decode** column, enter the name of the lookup you updated or created in step 3.
For example, **Lookup.DBUM.MySQL.UM.ReconTransformations**.
For trusted mode, use
Lookup.DBUM.MySQL.UM.ReconTransformations.Trusted.
10. Save the changes to the lookup definition.
11. Create a JAR with the class and upload it to the Oracle Identity Manager database as follows:

Run the Oracle Identity Manager Upload JARs utility to post the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

```
OIM_HOME/server/bin/UploadJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UploadJars.sh
```

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

12. Run the `PurgeCache` utility to clear content related to request datasets from the server cache.
13. Perform reconciliation to verify transformation of the field, for example, `SimpleDisplayName`.

5.7.7 Configuring Resource Exclusion Lists for MySQL

You can specify a list of accounts that must be excluded from reconciliation and provisioning operations. Accounts whose user IDs you specify in the exclusion list are not affected by reconciliation and provisioning operations.

In one of the lookup definitions for exclusion lists, enter the user IDs of target system accounts for which you do not want to perform provisioning and reconciliation operations. See [Lookup Definitions for Exclusion Lists for MySQL](#) for information about the lookup definitions and the format of the entries in these lookups.

To add entries in the lookup for exclusions during provisioning and reconciliation operations for MySQL:

1. On the Design Console, expand **Administration** and then double-click **Lookup Definition**.
2. Search for and open the **Lookup.DBUM.MySQL.UM.ExclusionList** lookup definition.
3. Click **Add**.
4. In the Code Key column, enter the resource object field name on which the exclusion list is applied. In the Decode column, enter the corresponding ID of the record to exclude.

For example, if you do not want to provision users with the user ID User001, then you must populate the lookup definition with the following values:

Code Key	Decode
User Name	User001

 **Note:**

If you want to specify a list of accounts that must be excluded during reconciliation or provisioning, the code key value being specified here must be exactly as the corresponding code key value in the `Lookup.DBUM.MySQL.UM.ReconAttrMap` lookup definition, or in the `Lookup.DBUM.MySQL.UM.ProvAttrMap` lookup definition, respectively.

5. If there is more than one user ID to exclude, then in the decode column, enter a list of all user IDs to exclude. Note that each User ID must be separated by a vertical bar (|).

For example, if you do not want to provision users with user IDs User001, User002, and User088 then you must populate the lookup definition with the following values:

Code Key	Decode
User Name	User001 User002 User088

You can also perform pattern matching to exclude user accounts. You can specify regular expressions supported by the representation in the `java.util.regex.Pattern` class.

 **See Also:**

For information about the supported patterns, visit <http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html>

For example, if you do not want to provision users matching any of the user IDs User001, User002, and User088, then you must populate the lookup definition with the following values:

Code Key	Decode
User Name[PATTERN]	User001 User002 User088

If you do not want to provision users whose user IDs start with 00012, then you must populate the lookup definition with the following values:

Code Key	Decode
User Name[PATTERN]	00012*

- Click the save icon.

5.7.8 Setting Up Action Scripts for MySQL

Learn about action scripts and how to configure them to run before or after the create, update, or delete an account provisioning operations.

This section provides information about the following topics:

- [About Action Scripts for MySQL](#)
- [Configuring Action Scripts for MySQL](#)

5.7.8.1 About Action Scripts for MySQL

Action scripts are scripts that you can configure to run before or after the create, update, or delete an account provisioning operations. For example, you could configure a script to run before every user creation. In another scenario, suppose you have a table called AUDIT_USERLOG where you want to log user creation activities performed only by the connector. Then, you could create and use after create script for adding data to this table after create operation.

Note:

To configure a before or after action, your connector must support running scripts. An exception is Groovy (with target set to **Connector**), which the Identity Connector Framework (ICF) supports by default for all converged connectors.

Every connector should specify which scripting language and which target it supports. This connector supports the following script:

- shell:** shell script
- target:** Connector

The target refers to the location where the script is executed. In this case, the script is executed on the same computer (JVM or .NET Runtime) where the connector is

deployed. For example, if you deploy the connector on the connector server, the script will be executed on that computer.

That is, if you are using a local framework, the script runs in your JVM. If you are connected to a remote framework, the script runs in the remote JVM or .NET Runtime.

5.7.8.2 Configuring Action Scripts for MySQL

To configure the action:

1. Log in to the Design Console.
2. Search for and open the **Lookup.DBUM.MySQL.UM.Configuration** lookup definition.
3. Add the following new values:

- **Code Key:** Before Create Action Language
- **Decode:** Enter the scripting language of the script you want to execute
- **Sample values:** SQL or STOREDPROC

4. Add these new values:

- **Code Key:** Before Create Action File
- **Decode:** Enter the full path to the file containing the script to be executed (Oracle Identity Manager must be able to access this file.)
- **Example:** /home/scripts/testscript.sql

This script may have a query as follows:

```
INSERT INTO AUDIT_USERLOG VALUES (['__NAME__'], CURRENT_TIMESTAMP))
```

5. Add these new values:

- **Code Key:** Before Create Action Target
- **Decode:** Connector

6. Save the lookup definition.

Now, this action will be executed every time you create a user. You must configure these three values for each action you want to execute.

6

Using and Extending the Connector for DB2

You can use the Database User Management Connector for DB2 for performing reconciliation and provisioning operations after configuring it to meet your requirements. You can also extend the functionality of the connector to address your specific business requirements. This chapter contains the following topics:

Note:

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

For Oracle Identity Manager hosted on a Microsoft Windows computer, if you have a previously installed connector, then you must extract the connector bundle zip file again before installing a new connector.

- [Configuring Secure Communication Between DB2 and Oracle Identity Manager](#)
- [Guidelines on Determining Values for the JDBC URL and Connection Properties Parameters for DB2](#)
- [Lookup Definitions for DB2](#)
- [Scheduled Jobs for DB2](#)
- [Reconciliation from DB2](#)
- [Provisioning for DB2](#)
- [Extending the Connector for DB2](#)

6.1 Configuring Secure Communication Between DB2 and Oracle Identity Manager

Note:

It is recommended that you perform the procedure described in this section to secure communication between the target system and Oracle Identity Manager.

Perform the following steps to configure secure communication between DB2 and Oracle Identity Manager:

1. See DB2 documentation for information about enabling SSL communication between DB2 and a client system. In this context, the client is Oracle Identity Manager.

Export the certificate on the DB2 host computer, and then restart the database service.

2. Copy the certificate to the Oracle Identity Manager host computer.
3. Import the certificate into the JVM truststore of the application server on which Oracle Identity Manager is running.

To import the certificate into the truststore, run the following command:

```
..\..\bin\keytool -import -file FILE_LOCATION -keystore TRUSTSTORE_LOCATION -storepass TRUSTSTORE_PASSWORD -trustcacerts -alias ALIAS
```

In this command:

- Replace *FILE_LOCATION* with the full path and name of the certificate file.
- Replace *ALIAS* with an alias for the certificate.
- Replace *TRUSTSTORE_PASSWORD* with a password for the truststore.
- Replace *TRUSTSTORE_LOCATION* with one of the truststore paths from [Table 6-1](#). This table shows the location of the truststore for each of the supported application servers.

Table 6-1 Truststore Locations on Supported Application Servers

Application Server	Truststore Location
Oracle WebLogic Server	<ul style="list-style-type: none"> • If you are using Oracle jrockit_R27.3.1-jdk, then import the certificate into the keystore in the following directory: <i>JROCKIT_HOME</i>/jre/lib/security • If you are using the default Oracle WebLogic Server JDK, then import the certificate into the keystore in following directory: <i>WEBLOGIC_HOME</i>/java/jre/lib/security/cacerts • If you are using a JDK other than Oracle jrockit_R27.3.1-jdk or Oracle WebLogic Server JDK, then import the certificate into your keystore at the following directory: <i>JAVA_HOME</i>/jre/lib/security/cacerts

4. To enable secure communication between DB2 and Oracle Identity Manager, set the value of the UseSSL IT resource parameter to `true`. You must provide a value for this parameter while performing the procedure described in [Configuring the IT Resource for the Connector Server](#).

6.2 Guidelines on Determining Values for the JDBC URL and Connection Properties Parameters for DB2

This section discusses the JDBC URL and Connection Properties parameters. You apply the information in this section while performing the procedure described in [Configuring the IT Resource for the Target System](#).

The following are guidelines on specifying the JDBC URL and Connection Properties parameters:

- **JDBC URL parameter**
Enter the following component of the connection URL as the value of the JDBC URL provider:

```
jdbc:db2://[SERVER_NAME][:PORT_NUMBER]/[DATABASE_NAME]
```

In this format:

- *SERVER_NAME* is the IP address (not the host name) of the target system host computer.
- *PORT_NUMBER* is the port at which the target system database is listening.
- *DATABASE_NAME* is the name of the database we are connecting.

The following is a sample value for the Database URL parameter:

```
jdbc:db2://192.168.16.76:50000/DBUSER
```

- **Connection Properties parameter**

Enter the following component of the connection URL as the value of the Connection Properties parameter:

```
[, PROPERTY=VALUE[, PROPERTY=VALUE]] . . .
```

In this format:

- *PROPERTY* is the name of one or more database connection properties, such as `applicationName` and `disableStatementPooling`.
- *VALUE* is the value of each database connection property whose name you specify by using the *PROPERTY* placeholder.

 **Note:**

Semicolons must be changed to number signs (#) in the value that you specify.

The following is a sample value for the Connection Properties parameter:

```
databaseName=sales#port=50000
```

6.3 Lookup Definitions for DB2

This section contains the following information:

- [Lookup Definitions Synchronized with DB2](#)
- [Lookup Definitions for Configurations for DB2](#)
- [Lookup Definitions for Attribute Mappings for DB2](#)
- [Lookup Definitions for Exclusion Lists for DB2](#)
- [Lookup Definitions for Transformation of Data in DB2](#)
- [Lookup Definition for Validation of Data in DB2](#)

6.3.1 Lookup Definitions Synchronized with DB2

During a provisioning operation, you use a lookup field on the process form to specify a single value from a set of values. For example, you use the Roles lookup field to select a role to be assigned to a user from the list of available roles. When you deploy the connector, lookup definitions corresponding to the lookup fields on the target system are created in

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

The connector provides predefined SQL queries for fetching values from the target system lookup fields into the lookup definitions in Oracle Identity Manager. These predefined SQL queries are stored in the `LoVSearch.queries` file with in the connector bundle.

After lookup definition synchronization, data is stored in the following format:

- Code Key value: `IT_RESOURCE_KEY~LOOKUP_FIELD_ID`
 In this format:
 - `IT_RESOURCE_KEY` is the numeric code assigned to each IT resource in Oracle Identity Manager.
 - `LOOKUP_FIELD_ID` is the target system code assigned to each lookup field entry.

Sample value: `1~SYS_ADM`

- Decode value: `IT_RESOURCE_NAME~LOOKUP_FIELD_ID`
 In this format:
 - `IT_RESOURCE_NAME` is the name of the IT resource in Oracle Identity Manager.
 - `LOOKUP_FIELD_ID` is the target system code assigned to each lookup field entry.

Sample value: `DB2 DB~SYS_ADM`

While performing a provisioning operation in Identity Self Service, you select the IT resource for the target system on which you want to perform the operation. When you perform this action, the lookup definitions on the page are automatically populated with values corresponding to the IT resource (target system installation) that you select. If your environment has multiple installations of the target system, then values corresponding to all IT resources are displayed.

[Table 6-2](#) lists column names of the tables in DB2 that are synchronized with their corresponding lookup definitions in Oracle Identity Manager.

Table 6-2 Lookup Definitions Synchronized with DB2

Lookup Definition	Target Column Name
Lookup.DBUM.DB2.Schema	Schema
Lookup.DBUM.DB2.Tablespace	Tablespace
Lookup.DBUM.DB2.UserType	User Type

By default, the `Lookup.DBUM.DB2.UserType` lookup definition contains the following entries for user types:

Code Key	Decode
GROUP	GROUP
USER	USER

The Lookup.DBUM.DB2.WithGrantOption lookup definition is used with tablespaces and schema. If you select With Grant Option, then tablespaces and schema are granted with this option. The Lookup.DBUM.DB2.WithGrantOption lookup definition contains the following entry:

Code Key	Decode
WITH GRANT OPTION	WITH GRANT OPTION

6.3.2 Lookup Definitions for Configurations for DB2

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

This section provides information about the following lookup definitions

- [Lookup.DBUM.DB2.Configuration](#)
- [Lookup.DBUM.DB2.UM.Configuration](#)
- [Lookup.DBUM.DB2.Configuration.Trusted](#)
- [Lookup.DBUM.DB2.UM.Configuration.Trusted](#)

6.3.2.1 Lookup.DBUM.DB2.Configuration

The Lookup.DBUM.DB2.Configuration lookup definition holds connector configuration entries that are used during target resource reconciliation and provisioning operations.

Table 6-3 Entries in Lookup.DBUM.DB2.Configuration

Code Key	Decode Key	Description
Bundle Name	org.identityconnectors.dbum	Name of the connector bundle package Do not modify this entry.
Bundle Version	1.0.1116	Version of the connector bundle class Do not modify this entry.
Connector Name	org.identityconnectors.dbum.DBUMConnector	Name of the connector class Do not modify this entry.
disableValuesSet	"NN", "YN", "NY"	Possible values for the disabled status of a user
reservedWordsList	"GRANT", "REVOKE", "OF", "ON", "TO", "DATABASE", "TABLESPACE", "SCHEMA", "CREATEIN", "ALTERIN", "DROPIN", "FROM", "USE"	List of words that are reserved and are not allowed to be used in the names of the connector artifacts
User Configuration Lookup	Lookup.DBUM.DB2.UM.Configuration	Name of the lookup definition that contains user-specific configuration properties Do not modify this entry.

6.3.2.2 Lookup.DBUM.DB2.UM.Configuration

The Lookup.DBUM.DB2.UM.Configuration lookup definition holds user-specific connector configuration entries that are used during target resource reconciliation and provisioning operations.

Table 6-4 Entries in Lookup.DBUM.DB2.UM.Configuration

Code Key	Decode Key
Provisioning Attribute Map	Lookup.DBUM.DB2.UM.ProvAttrMap
Provisioning Exclusion List	Lookup.DBUM.DB2.UM.ExclusionList
Provisioning Validation Lookup	Lookup.DBUM.DB2.UM.ProvValidations
Recon Validation Lookup	Lookup.DBUM.DB2.UM.ReconValidations
Recon Attribute Map	Lookup.DBUM.DB2.UM.ReconAttrMap
Recon Exclusion List	Lookup.DBUM.DB2.UM.ExclusionList
Recon Transformation Lookup	Lookup.DBUM.DB2.UM.ReconTransformations

6.3.2.3 Lookup.DBUM.DB2.Configuration.Trusted

The Lookup.DBUM.DB2.Configuration.Trusted lookup definition holds connector configuration entries that are used during reconciliation and provisioning operations in trusted source mode.

Table 6-5 Entries in Lookup.DBUM.DB2.Configuration.Trusted

Code Key	Decode Key
Bundle Name	org.identityconnectors.dbum
Bundle Version	1.0.1116
Connector Name	org.identityconnectors.dbum.DBUMConnector
disableValuesSet	"LOCKED"
User Configuration Lookup	Lookup.DBUM.DB2.UM.Configuration.Trusted

6.3.2.4 Lookup.DBUM.DB2.UM.Configuration.Trusted

The Lookup.DBUM.DB2.UM.Configuration.Trusted lookup definition holds user-specific connector configuration entries that are used during reconciliation and provisioning operations in trusted source mode.

Table 6-6 Entries in Lookup.DBUM.DB2.UM.Configuration.Trusted

Code Key	Decode Key
Recon Attribute Defaults	Lookup.DBUM.DB2.UM.ReconDefaults.Trusted
Recon Attribute Map	Lookup.DBUM.DB2.UM.ReconAttrMap.Trusted

Table 6-6 (Cont.) Entries in Lookup.DBUM.DB2.UM.Configuration.Trusted

Code Key	Decode Key
Recon Exclusion List	Lookup.DBUM.DB2.UM.ExclusionList.Trusted
Recon Transformation Lookup	Lookup.DBUM.DB2.UM.ReconTransformations.Trusted

6.3.3 Lookup Definitions for Attribute Mappings for DB2

This section describes the following lookup definitions:

- [Lookup.DBUM.DB2.UM.ProvAttrMap](#)
- [Lookup.DBUM.DB2.UM.ReconAttrMap](#)
- [Lookup.DBUM.DB2.UM.ReconAttrMap.Trusted](#)
- [Lookup.DBUM.DB2.UM.ReconDefaults.Trusted](#)

6.3.3.1 Lookup.DBUM.DB2.UM.ProvAttrMap

The Lookup.DBUM.DB2.UM.ProvAttrMap lookup definition holds user-specific mappings between process form fields (Code Key values) and target system attributes (Decode values) used during provisioning operations.

Table 6-7 Entries in Lookup.DBUM.DB2.UM.ProvAttrMap

Code Key	Decode Key
Return Id	__UID__
UD_DB_DB2_S~Schema[LOOKUP]	schemas~DBSchema~__NAME__
UD_DB_DB2_S~Schema Grant Option	schemas~DBSchema~grantOption
UD_DB_DB2_T~Tablespace[LOOKUP]	tablespaces~DBTablespaces~__NAME__
UD_DB_DB2_T~Tablespace Grant Option	tablespaces~DBTablespaces~grantOption
Username	__NAME__
User Type	userType

6.3.3.2 Lookup.DBUM.DB2.UM.ReconAttrMap

The Lookup.DBUM.DB2.UM.ReconAttrMap lookup definition holds user-specific mappings between reconciliation attribute names as specified in the resource object (Code Key values) and target system attributes (Decode values) used during reconciliation operations.

Table 6-8 Entries in Lookup.DBUM.DB2.UM.ReconAttrMap

Code Key	Decode Key
Return ID	__UID__
Schema List~Schema Grant option	schemas~DBSchema~grantOption
Schema List~Schema Name[LOOKUP]	schemas~DBSchema~__NAME__
Status	__ENABLE__
Tablespace List~Tablespace Grant Option	tablespaces~DBTablespaces~grantOption
Tablespace List~Tablespace Name[LOOKUP]	tablespaces~DBTablespaces~__NAME__
User Name	__UID__
User Type	userType

6.3.3.3 Lookup.DBUM.DB2.UM.ReconAttrMap.Trusted

The Lookup.DBUM.DB2.UM.ReconAttrMap.Trusted lookup definition holds user-specific mappings between reconciliation attribute names as specified in the resource object (Code Key values) and target system attributes (Decode values) used during reconciliation operations in trusted source mode.

Table 6-9 Entries in Lookup.DBUM.DB2.UM.ReconAttrMap.Trusted

Code Key	Decode Key
First Name	__UID__
User ID	__UID__

6.3.3.4 Lookup.DBUM.DB2.UM.ReconDefaults.Trusted

This lookup definition contains the default values for the Oracle Identity Manager user attributes. You can change these values as per your requirements.

For example, if you want the users reconciled from a trusted source to be part of the MyORG organization, then map the lookup definition entry as follows:

Code Key = Organization Name

Decode = MyORG (instead of Xellerate Users)

Table 6-10 Entries in Lookup.DBUM.DB2.UM.ReconDefaults.Trusted

Code Key	Decode Key
Empl Type	Full-Time
Organization Name	Xellerate Users
Status	Active

Table 6-10 (Cont.) Entries in Lookup.DBUM.DB2.UM.ReconDefaults.Trusted

Code Key	Decode Key
User Type	End-User

6.3.4 Lookup Definitions for Exclusion Lists for DB2

This section describes the lookup definitions that hold resources for which you do not want to perform provisioning and reconciliation operations. Exclusions can be applied to any attribute in the process form or reconciliation profile. The Code Key value must be one of the Code Key values in Lookup.DBUM.DB2.UM.ReconAttrMap or Lookup.DBUM.DB2.UM.ProvAttrMap lookup definitions.

Depending on how the target system is configured, you can use one of the following lookups:

- For target resource mode: Lookup.DBUM.DB2.UM.ExclusionList

By default, this lookup definition has the following entry:

Code Key	Decode
User Name	db2admin

- For trusted source mode: Lookup.DBUM.DB2.UM.ExclusionList.Trusted

By default, this lookup definition has the following entry:

Code Key	Decode
User ID	db2admin

The following is the format of the values stored in these lookups:

Code Key	Decode	Sample Values
User Name	User ID of a user	Code Key: User Name Decode: User001
User Name with the [PATTERN] suffix	A regular expression supported by the representation in the <code>java.util.regex.Pattern</code> class	Code Key: User Name[PATTERN] To exclude users matching any of the user ID 's User001, User002, User088, then: Decode: User001 User002 User088 To exclude users whose user ID 's start with 00012, then: Decode: 00012* See Also: For information about the supported patterns, visit http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html

[Configuring Resource Exclusion Lists for DB2](#) describes the procedure to add entries in these lookup definitions.

6.3.5 Lookup Definitions for Transformation of Data in DB2

This section describes the lookup definitions that hold resources for which you want to enable transformation of data during reconciliation operations.

Depending on how the target system is configured, use one of the following lookup definitions:

- For target resource mode: Lookup.DBUM.DB2.UM.ReconTransformations

By default, this lookup definition has the following entries:

Code Key	Decode
Schema List	db2.iam.connectors.dbum.transformations.SchemaGrantOptionTransformation
Tablespace List	db2.iam.connectors.dbum.transformations.TablespaceGrantOptionTransformation
User Type	db2.iam.connectors.dbum.transformations.UserTypeTransformation

- For trusted source mode: Lookup.DBUM.DB2.UM.ReconTransformations.Trusted

[Configuring Transformation of Data During User Reconciliation for DB2](#) describes the procedure to add entries in these lookup definitions.

6.3.6 Lookup Definition for Validation of Data in DB2

You can use the Lookup.DBUM.DB2.UM.ProvValidations lookup to configure validation of data during provisioning operations.

[Configuring Validation of Data During Reconciliation and Provisioning for DB2](#) describes the procedure to add entries in this lookup definition.

6.4 Scheduled Jobs for DB2

When you run the Connector Installer or import the connector XML file, the scheduled jobs are automatically created in Oracle Identity Manager.

This section describes the following topics:

- [Scheduled Jobs for Lookup Field Synchronization for DB2](#)
- [Attributes for Scheduled Jobs for DB2](#)
- [Configuring Scheduled Jobs for DB2](#)

6.4.1 Scheduled Jobs for Lookup Field Synchronization for DB2

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

The following scheduled jobs are used for lookup field synchronization:

- DBUM DB2 Schema Lookup Reconciliation
- DBUM DB2 Tablespaces Lookup Reconciliation

You must specify values for the attributes of these scheduled jobs. [Table 6-11](#) describes the attributes of these scheduled jobs. The procedure to configure scheduled jobs is described later in the guide.

Table 6-11 Attributes of the Scheduled Jobs for Lookup Field Synchronization

Attribute	Description
Code Key Attribute	Enter the name of the connector or target system attribute that is used to populate the Code Key column of the lookup definition (specified as the value of the Lookup Name attribute). Sample value: <code>__NAME__</code> Note: Do not change the value of this attribute.
Decode Attribute	Enter the name of the connector or target system attribute that is used to populate the Decode column of the lookup definition (specified as the value of the Lookup Name attribute). Sample value: <code>__NAME__</code>
IT Resource Name	Enter the name of the IT resource for the target system installation from which you want to reconcile user records. Default value: <code>DB2</code>
Lookup Name	This attribute holds the name of the lookup definition that maps each lookup definition with the data source from which values must be fetched. Depending on the scheduled job you are using, the default values are as follows: <ul style="list-style-type: none"> For DBUM DB2 Schema Lookup Reconciliation - <code>Lookup.DBUM.DB2.Schema</code> For DBUM DB2 Tablespaces Lookup Reconciliation - <code>Lookup.DBUM.DB2.Tablespaces</code>
Object Type	Enter the type of object whose values must be synchronized. Depending on the scheduled job you are using, the default values are as follows: <ul style="list-style-type: none"> For DBUM DB2 Schema Lookup Reconciliation - <code>__SCHEMAS__</code> For DBUM DB2 Tablespaces Lookup Reconciliation - <code>__TABLESPACES__</code> Note: Do not change the value of this attribute.
Resource Object Name	Enter the name of the resource object that is used for reconciliation. Default value: <code>DB2 DB User</code>

6.4.2 Attributes for Scheduled Jobs for DB2

The following scheduled job is used to reconcile user data in the target resource (account management) mode of the connector:

- DBUM DB2 User Target Reconciliation

The following scheduled job is used to reconcile user data in the trusted source (identity management) mode of the connector:

- DBUM DB2 User Trusted Reconciliation

[Table 6-12](#) describes the attributes of the scheduled jobs.

Table 6-12 Attributes of the Scheduled Jobs for Reconciliation

Attribute	Description
Batch Size	Value for running the scheduled job in batch mode. By default, this value is empty.
Filter	Expression for filtering records that must be reconciled by the scheduled job By default, the value of this attribute is empty. Sample value: <code>equalTo('__UID__', 'SEPT12USER1')</code> See Performing Limited Reconciliation from DB2 for the syntax of this expression.
IT Resource Name	Name of the IT resource for the target system installation from which you want to reconcile user records For DBUM DB2 User Target Reconciliation: DB2 For DBUM DB2 User Trusted Reconciliation, enter the name of the IT resource created for trusted source mode.
Object Type	Type of object you want to reconcile Default value: <code>User</code>
Resource Object Name	Name of the resource object that is used for reconciliation For DBUM DB2 User Target Reconciliation: <code>DB2 DB User</code> For DBUM DB2 User Trusted Reconciliation: <code>DB2 DB Trusted</code>
Scheduled Task Name	Name of the scheduled job Note: For the scheduled job included with this connector, you must not change the value of this attribute. However, if you create a copy of the task, then you can enter the unique name for that scheduled job as the value of this attribute.

6.4.3 Configuring Scheduled Jobs for DB2

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

See [Scheduled Jobs for Lookup Field Synchronization for DB2](#) and [Attributes for Scheduled Jobs for DB2](#) for the scheduled jobs that are part of the connector and for information about their attributes.

To configure a scheduled job:

1. If you are using Oracle Identity Manager release 11.1.1.x:
 - a. Log in to the Administrative and User Console.
 - b. On the Welcome to Oracle Identity Manager Self Service page, click **Advanced** in the upper-right corner of the page.
 - c. On the Welcome to Oracle Identity Manager Advanced Administration page, in the System Management region, click **Search Scheduled Jobs**.
2. If you are using Oracle Identity Manager release 11.1.2.x or later:
 - a. Log in to Oracle Identity System Administration.
 - b. In the left pane, under System Management, click **Scheduler**.
3. Search for and open the scheduled job as follows:

- a. On the left pane, in the Search field, enter the name of the scheduled job as the search criterion. Alternatively, you can click **Advanced Search** and specify the search criterion.
 - b. In the search results table on the left pane, click the scheduled job in the Job Name column.
4. On the Job Details tab, you can modify the following parameters:

Retries: Enter an integer value in this field. This number represents the number of times the scheduler tries to start the job before assigning the Stopped status to the job.

Schedule Type: Depending on the frequency at which you want the job to run, select the appropriate schedule type.

 **Note:**

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

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

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

 **Note:**

- Attribute values are predefined in the connector XML file that you import. Specify values only for those attributes that you want to change.
- Attributes of the scheduled job are discussed in [Attributes for Scheduled Jobs for DB2](#).

6. After specifying the attributes, click **Apply** to save the changes.

6.5 Reconciliation from DB2

Postinstallation steps are divided across the following sections:

As mentioned earlier in this guide, reconciliation involves duplicating in Oracle Identity Manager the creation of and modifications to user accounts on the target system. This section discusses the following topics related to configuring reconciliation:

- [Guidelines on Configuring Reconciliation for DB2](#)
- [About the Reconciliation Process for DB2](#)
- [Target System Columns Used in Reconciliation from DB2](#)
- [Configuring the Target System As a Trusted Source](#)
- [Reconciliation Rules for DB2](#)
- [Viewing Reconciliation Rules for DB2](#)
- [Reconciliation Action Rules for DB2](#)

- [Viewing Reconciliation Action Rules for DB2](#)
- [Performing Full Reconciliation from DB2](#)
- [Performing Limited Reconciliation from DB2](#)
- [Performing Batched Reconciliation from DB2](#)

6.5.1 Guidelines on Configuring Reconciliation for DB2

The following are guidelines that you must apply while configuring reconciliation:

- Before a target resource reconciliation run is performed, lookup definitions must be synchronized with the lookup fields of the target system. In other words, the scheduled job for lookup field synchronization must be run before user reconciliation runs.
- After you configure batched reconciliation, if reconciliation fails during a batched reconciliation run, then rerun the scheduled job without changing the values of the task attributes.

6.5.2 About the Reconciliation Process for DB2

This connector can be configured to perform either trusted source reconciliation or target resource reconciliation.

When you configure the target system as a target resource, the connector enables you to create and manage database accounts for OIM Users through provisioning. In addition, data related to newly created and modified target system accounts can be reconciled and linked with existing OIM Users and provisioned resources.

When you configure the target system as a trusted source, the connector fetches into Oracle Identity Manager, data about newly created target system accounts. This data is used to create OIM Users. See [Configuring the Target System As a Trusted Source](#) for more information.



See Also:

Reconciliation Based on the Object Being Reconciled in *Oracle Fusion Middleware Administering Oracle Identity Manager* for conceptual information about target resource reconciliation and trusted source reconciliation.

The following is an overview of the steps involved in reconciliation:

1. A SQL query or stored procedure is used to fetch target system records during reconciliation.
2. The scheduled job communicates to connector bundle and runs search operations over it, maps the task attributes to parameters of the reconciliation query or stored procedure, and then runs the query or stored procedure on the target system.
3. Target system records that meet the query or stored procedure criteria are fetched into Oracle Identity Manager.

4. If you have configured your target system as a trusted source, then each user record fetched from the target system is compared with existing OIM Users. The reconciliation rule is applied during the comparison process.

The next step of the process depends on the outcome of the matching operation:

- If a match is found between the target system record and the OIM User, then the OIM User attributes are updated with changes made to the target system record.
- If no match is found, then the target system record is used to create an OIM User.

5. If you have configured your target system as a target resource, then each user record fetched from the target system is compared with existing target system resources assigned to OIM Users. The reconciliation rule is applied during the comparison process.

The next step of the process depends on the outcome of the matching operation:

- If a match is found between the target system record and a resource provisioned to an OIM User, then the database user resource is updated with changes made to the target system record.
- If no match is found, then the target system user record is compared with existing OIM Users. The next step depends on the outcome of the matching operation:

If a match is found, then the target system record is used to provision a resource for the OIM User.

If no match is found, then the status of the reconciliation event is set to No Match Found.

**Note:**

[Reconciliation Rules for DB2](#) for information about the reconciliation rule

6.5.3 Target System Columns Used in Reconciliation from DB2

As mentioned earlier in this guide, this connector can be configured to perform either target resource reconciliation or trusted source reconciliation. This section discusses the following topics:

- The `Lookup.DBUM.DB2.UM.ReconAttrMap` lookup definition holds attribute mappings for user reconciliation. This lookup definition contains mapping of Oracle Identity Manager attributes and connector attributes.

See [Lookup.DBUM.DB2.UM.ReconAttrMap](#) for more information.

- The `Lookup.DBUM.DB2.UM.ReconAttrMap.Trusted` lookup definition holds attribute mappings for reconciliation in trusted mode. This lookup definition maps reconciliation profile attributes and connector attributes used in the reconciliation query. In addition, the connector attributes are associated to columns within the bundle.

See [Lookup.DBUM.DB2.UM.ReconAttrMap.Trusted](#) for more information about this lookup definition.

6.5.4 Configuring the Target System As a Trusted Source



Note:

Skip this section if you do not want to designate the target system as a trusted source for reconciliation.

To configure trusted source reconciliation:

1. If you are using Oracle Identity Manager release 11.1.1.x:
 - a. Log in to the Administrative and User Console.
 - b. On the Welcome page, click **Advanced** in the upper-right corner of the page.
 - c. On the Welcome to Oracle Identity Manager Advanced Administration page, in the Configuration region, click **Create IT Resource**.
 2. If you are using Oracle Identity Manager release 11.1.2.x or later:
 - a. Log in to Oracle Identity System Administration.
 - b. In the left pane, under Configuration, click **IT Resource**.
 - c. In the Manage IT Resource page, click **Create IT Resource**.
 3. On the Step 1: Provide IT Resource Information page, enter the following information:
 - **IT Resource Name:** Enter a name for the IT resource. For example, DB2 DB Trusted.
 - **IT Resource Type:** Select the **DB2 DB** IT resource type for the IT resource.
 4. Click **Continue**.
 5. On the Step 2: Specify IT Resource Parameter Values page, specify values for the parameters of the IT resource.

Configuration Lookup: Name of the lookup definition in which you store the connector configuration information for the target system.

Sample Value: `Lookup.DBUM.DB2.Configuration.Trusted`

Provide values for the other IT resource parameters.
 6. Click **Continue**.
- In the following steps, provide permissions on the IT resource that you are creating as per your requirements.

You can use this IT resource for trusted source reconciliation operations.

6.5.5 Reconciliation Rules for DB2

See Also:

Reconciliation Metadata in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for generic information about reconciliation rules and reconciliation action rules

This section describes the reconciliation rules used by the reconciliation engine for this connector.

The following are the reconciliation rules for target resource reconciliation:

- **Rule name:** DBUM DB2 Target Recon
- **Rule element:** User Login Equals User Name

The following are the reconciliation rules for trusted source reconciliation:

- **Rule name:** DB2 DB Trusted
- **Rule element:** User Login Equal User ID

In these rule elements:

- User Login is the field on the OIM User form.
- User Name and User ID are the target system fields.

6.5.6 Viewing Reconciliation Rules for DB2

After you deploy the connector, you can view the reconciliation rule for reconciliation by performing the following steps:

Note:

Perform the following procedure only after the connector is deployed.

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Development Tools**.
3. Double-click **Reconciliation Rules**.
4. Search for the rule name.

6.5.7 Reconciliation Action Rules for DB2

Reconciliation action rules define that actions the connector must perform based on the reconciliation rules defined for Users.

[Table 6-13](#) lists the action rules for target resource reconciliation.

Table 6-13 Action Rules for Target Resource Reconciliation

Rule Condition	Action
No Matches Found	Assign to Administrator With Least Load
One Entity Match Found	Establish Link
One Process Match Found	Establish Link

[Table 6-14](#) lists the action rules for trusted source reconciliation.

Table 6-14 Action Rules for Trusted Source Reconciliation

Rule Condition	Action
No Matches Found	Create User
One Entity Match Found	Establish Link

6.5.8 Viewing Reconciliation Action Rules for DB2

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

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Resource Management**.
3. Double-click **Resource Objects**.
4. Search for and open the resource object. The following are the names of the resource objects for each target system database:
 - Resource object for DB2:
DB2 DB User
 - Resource object for DB2 as trusted source:
DB2 DB Trusted
5. Click the **Object Reconciliation** tab, and then click the **Reconciliation Action Rules** tab. The Reconciliation Action Rules tab displays the action rules defined for this connector.

6.5.9 Performing Full Reconciliation from DB2

Full reconciliation involves reconciling all existing user records from the target system into Oracle Identity Manager. After you deploy the connector, you must first perform full reconciliation.

To perform a full reconciliation run, remove (delete) any value currently assigned to the Filter attribute and run one of the following scheduled jobs:

- For DB2 as a target resource: DBUM DB2 User Target Reconciliation
- For DB2 as a trusted source: DBUM DB2 User Trusted Reconciliation

See [Attributes for Scheduled Jobs for DB2](#) for information about this scheduled job.

6.5.10 Performing Limited Reconciliation from DB2

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

You can perform limited reconciliation by creating filters for the reconciliation module. This connector provides a Filter attribute (a scheduled task attribute) that allows you to use any of the DBUM resource attributes to filter the target system records. You can apply filters to the parent parameters in the reconciliation query file stored in a JAR file in the bundle directory of the connector installation media. For example, to locate the reconciliation query file, you can extract the `bundle/org.identityconnectors.dbum-1.0.1116.jar` file and open `scripts/db2/Search.queries`.

The following table provides a list of parent parameters that can be used with the Filter attribute of the scheduled jobs:

Parameter	Description
<code>__UID__</code>	Unique identity representing the user This parameter is mapped to USERNAME or <code>__NAME__</code> connector attribute.
<code>userType</code>	Type of the user account The value of this parameter can be one of the following: USER or GROUP

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

While deploying the connector, follow the instructions in [Configuring Scheduled Jobs for DB2](#) to specify attribute values.

6.5.11 Performing Batched Reconciliation from DB2

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

You can configure batched reconciliation to avoid these problems.

To configure batched reconciliation, you must specify value for the Batch Size reconciliation scheduled job attribute. Use this attribute to specify the number of records that must be included in each batch. By default, this value is empty.

If you specify a value other than `All`, then some of the newly added or modified user records may not get reconciled during the current reconciliation run. The following example illustrates this:

Suppose you specify the Batch Size value as `200` while configuring the scheduled jobs. Suppose that 314 user records were created or modified after the last reconciliation run. Of these 314 records, only 200 records would be reconciled during the current reconciliation run. The remaining 114 records would be reconciled during the next reconciliation run.

You specify values for the Batch Size attribute by following the instructions described in [Configuring Scheduled Jobs for DB2](#).

6.6 Provisioning for DB2

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

This section contains the following topics about provisioning:

- [Guidelines on Performing Provisioning Operations for DB2](#)
- [Understanding the Provisioning Process for DB2](#)
- [Configuring Direct Provisioning for DB2](#)
- [Configuring Request-Based Provisioning for DB2](#)
- [Enabling Request-Based Provisioning](#)
- [Switching Between Request-Based Provisioning and Direct Provisioning for DB2](#)
- [Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x](#)

6.6.1 Guidelines on Performing Provisioning Operations for DB2

The following are guidelines that you must apply while performing provisioning operations:

- Before you perform provisioning operations, lookup definitions must be synchronized with the lookup fields of the target system. In other words, run the scheduled jobs for lookup field synchronization before provisioning operations.
- DB2 does not allow deletion of created user accounts. Therefore, as part of the Revoke Resource operation of Oracle Identity Manager, the following changes will be made:
 - On the target system, the corresponding user account is set to Inactive, which revokes connect and database administration authorizations for the user.
 - In Oracle Identity Manager, the tasks for the corresponding account are cancelled and the account status is set to Disabled.
- Passwords for user accounts provisioned from Oracle Identity Manager must adhere to the password policy set in the target system.
- The character length of target system fields must be taken into account when specifying values for the corresponding Oracle Identity Manager fields.
- During an update password provisioning operation, ensure that you clear the existing text in the Password field, and then enter the new password.

6.6.2 Understanding the Provisioning Process for DB2

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

When you install the connector on Oracle Identity Manager, the direct provisioning feature is automatically enabled. This means that the process form is enabled when you install the connector.

The following are types of provisioning operations:

- Direct provisioning
- Request-based provisioning
- Provisioning triggered by policy changes

If you configure the connector for request-based provisioning, then the process form is suppressed and the object form is displayed. In other words, direct provisioning is disabled when you configure the connector for request-based provisioning. If you want to revert to direct provisioning, then see [Switching Between Request-Based Provisioning and Direct Provisioning for DB2](#).

The following is an overview of the Create User provisioning process in DB2 that is started through direct provisioning:

1. On the Create User page of the Administrative and User Console, the administrator enters the data required for an OIM User account creation.

Suppose the administrator enters the following values for the fields on the Create User page:

- First Name: John
- Last Name: Doe
- User ID: jdoe

An OIM User account is created for John Doe.

2. The administrator selects the resource to be provisioned to the OIM User account that has been created. In this example, the administrator selects the DB2 DB User resource.
3. The administrator enters the data required for provisioning the DB2 DB User resource. Suppose the administrator wants to create a local user that requires a password to log in to the database. Therefore, the administrator enters the following values on the resource provisioning process form:

- IT Resource: DB2
- Username: JDoe
- User Type: USER

In addition, the administrator also enters the following values on the process form for granting tablespace and schema:

- Tablespace: 3~USERSPACE1
- Tablespace Grant Option: WITH GRANT OPTION
- Schema: 3~SYSTOOLSPACE
- Schema Grant Option: WITH GRANT OPTION

4. From the information available in the IT resource for the target system, the configuration (Lookup.DBUM.DB2.Configuration) lookup definition is identified. This lookup definition stores configuration information that is used during connector operations.
5. The connector bundle contains the script (Provisioning.queries) required for provisioning operations.

6. The identifiers in the SQL statement are replaced with the input parameters fetched from the query. Then, the SQL statement with actual values is formed.
7. The connector runs the SQL statement on DB2 and creates the jdoe account on the target system. The next step of the process depends on whether the administrator had entered data for granting tablespace or schema to the target system account.

If the administrator did not enter any values for granting tablespace or schema, then the provisioning process ends here. Otherwise, the process continues to the next step.
8. While performing Step 3, the administrator had entered the required data for granting tablespace and schema to the jdoe account. Therefore, the corresponding query as mentioned in Step 6 is read.
9. The complete SQL statement that must be run to perform the Add tablespace and schema provisioning operation is formed. Depending on whether the administrator had granted a tablespace with the grant option, the SQL statement is one of the following:
 - If the administrator specified a value for granting the tablespace With Grant Option, then the following SQL statement is formed:

```
GRANT USE OF TABLESPACE USERSPACE1 TO USER jdoe WITH GRANT OPTION
```
 - If the administrator did not specify a value for granting tablespace With Grant Option, then the following SQL statement is formed:

```
GRANT USE OF TABLESPACE USERSPACE1 TO USER jdoe
```
10. The input parameters required to run the SQL statement are fetched from the parameter configuration done using the queries in the query files.
11. The identifiers in the SQL statement (formed in Step 11) are replaced with the input parameters fetched from the query. Then, the SQL statement with actual values is formed.
12. The query runs the SQL statement on the target system (DB2) and grants the USERSPACE1 tablespace to the jdoe target system account.

6.6.3 Configuring Direct Provisioning for DB2

In direct provisioning, the Oracle Identity Manager administrator uses the Administrative and User Console to create a target system account for a user.

To provision a resource by using the direct provisioning approach:

1. Log in to the Administrative and User Console.
2. To first create an OIM User before provisioning a database account to the user:
 - a. On the Welcome to Identity Administration page, in the Users region, click **Create User**.
 - b. On the Create User page, enter values for the OIM User fields, and then click the save icon.
3. To search for an existing OIM User to be provisioned:
 - a. On the Welcome to Identity Administration page, search for the user by selecting **Users** from the Search list on the left pane.

 **Note:**

Direct provisioning allows the provisioning of multiple database accounts on the target system.

- Direct provisioning cannot be used if you enable request-based provisioning.

6.6.4.2 Enabling Request-Based Provisioning

The following sections provide information about the procedures you must perform to enable request-based provisioning:

 **Note:**

The procedure described in this section is applicable only if you are using Oracle Identity Manager release 11.1.1.x.

The following sections discuss the steps to be performed to enable request-based provisioning:

- [Approver's Role in Request-Based Provisioning for DB2](#)
- [Importing DB2 Request Datasets Using Deployment Manager](#)
- [End User's Role in Request-Based Provisioning for DB2](#)
- [Enabling the Auto Save Form Feature for DB2](#)
- [Running the PurgeCache Utility for DB2](#)

6.6.4.2.1 Approver's Role in Request-Based Provisioning for DB2

The following are steps performed by the approver in a request-based provisioning operation:

1. Log in to the Administrative and User Console.
2. On the Welcome page, click **Self-Service** in the upper-right corner of the page.
3. On the Welcome to Identity Manager Self Service page, click the **Tasks** tab.
4. On the **Approvals** tab, in the first section, you can specify a search criterion for request task that is assigned to you.
5. From the search results table, select the row containing the request you want to approve, and then click **Approve Task**.

A message confirming that the task was approved is displayed.

6.6.4.2.2 Importing DB2 Request Datasets Using Deployment Manager

A request dataset is an XML file that specifies the information to be submitted by the requester during a provisioning operation. These request datasets specify information about the default set of attributes for which the requester must submit information during a request-based provisioning operation.

To import a request dataset XML file by using the Deployment Manager:

1. Log in to the Oracle Identity Manager Administrative and User Console.
2. Click the **Deployment Management** link on the left navigation bar.
3. Click the **Import** link under Deployment Management.
A dialog box for opening files is displayed.
4. Locate and open the request dataset XML file, DBUserManagement-DB2-Datasets.xml, which is in the xml directory of the installation media.
Details of this XML file are shown on the **File Preview** page.
5. Click **Add File**.
The Substitutions page is displayed.
6. Click **Next**.
The Confirmation page is displayed.
7. Click **Import**.
8. Close the Deployment Manager dialog box.
The request dataset is imported into Oracle Identity Manager.

6.6.4.2.3 End User's Role in Request-Based Provisioning for DB2

The following steps are performed by the end user in a request-based provisioning operation:

1. Log in to the Administrative and User Console.
2. On the Welcome page, click **Advanced** in the upper-right corner of the page.
3. On the Welcome to Identity Administration page, click the **Administration** tab, and then click the **Requests** tab.
4. From the Actions menu on the left pane, select **Create Request**.
The Select Request Template page is displayed.
5. From the Request Template list, select **Provision Resource** and click **Next**.
6. On the Select Users page, specify a search criterion in the fields to search for the user that you want to provision the resource, and then click **Search**. A list of users that match the search criterion you specify is displayed in the Available Users list.
7. From the **Available Users** list, select the user to whom you want to provision the account.
If you want to create a provisioning request for more than one user, then from the **Available Users** list, select users to whom you want to provision the account.
8. Click **Move** or **Move All** to include your selection in the Selected Users list, and then click **Next**.
9. On the Select Resources page, click the arrow button next to the Resource Name field to display the list of all available resources.
10. From the Available Resources list, select **DB2 DB User**, move it to the Selected Resources list, and then click **Next**.
11. On the Resource Details page, enter details of the account that must be created on the target system, and then click **Next**.

12. On the Justification page, you can specify values for the following fields, and then click **Finish**.
 - Effective Date
 - JustificationA message confirming that your request has been sent successfully is displayed along with the Request ID.
13. If you click the request ID, then the Request Details page is displayed.
14. To view details of the approval, on the Request Details page, click the **Request History** tab.

6.6.4.2.4 Enabling the Auto Save Form Feature for DB2

To enable the Auto Save Form feature:

1. Log in to the Design Console.
2. Expand **Process Management**, and then double-click **Process Definition**.
3. Search for and open the **DB2 DB** process definition.
4. Select the **Auto Save Form** check box.
5. Click the save icon.

6.6.4.2.5 Running the PurgeCache Utility for DB2

Run the PurgeCache utility to clear content belonging to the Metadata category from the server cache.

See [Clearing Content Related to the Connector Resource Bundles from the Server Cache](#) for instructions.

The procedure to enable enabling request-based provisioning ends with this step.

6.6.5 Switching Between Request-Based Provisioning and Direct Provisioning for DB2

If you have configured the connector for request-based provisioning, you can always switch to direct provisioning. Similarly, you can always switch back to request-based provisioning any time. This section discusses the following topics:

- [Switching From Request-Based Provisioning to Direct Provisioning](#)
- [Switching From Direct Provisioning to Request-Based Provisioning](#)

6.6.5.1 Switching From Request-Based Provisioning to Direct Provisioning



Note:

It is assumed that you have performed the procedure described in [Configuring Request-Based Provisioning for DB2](#).

To switch from request-based provisioning to direct provisioning:

1. Log in to the Design Console.
2. Disable the Auto Save Form feature as follows:
 - a. Expand **Process Management**, and then double-click **Process Definition**.
 - b. Search for and open the **DB2 DB** process definition.
 - c. Deselect the Auto Save Form check box.
 - d. Click the save icon.
3. If the Self Request Allowed feature is enabled, then:
 - a. Expand **Resource Management**, and then double-click **Resource Objects**.
 - b. Search for and open the **DB2 DB User** resource object.
 - c. Deselect the Self Request Allowed check box.
 - d. Click the save icon.

6.6.5.2 Switching From Direct Provisioning to Request-Based Provisioning

To switch from direct provisioning back to request-based provisioning:

1. Log in to the Design Console.
2. Enable the Auto Save Form feature as follows:
 - a. Expand **Process Management**, and then double-click **Process Definition**.
 - b. Search for and open the **DB2 DB** process definition.
 - c. Select the **Auto Save Form** check box.
 - d. Click the save icon.
3. If you want to enable end users to raise requests for themselves, then:
 - a. Expand **Resource Management**, and then double-click **Resource Objects**.
 - b. Search for and open the **DB2 DB User** resource object.
 - c. Select the Self Request Allowed check box.
 - d. Click the save icon.

6.6.6 Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x

To perform provisioning operations in Oracle Identity Manager release 11.1.2.x:

1. Log in to Oracle Identity Self Service.
2. If you want to first create an OIM User and then provision a target system account, then:

 **Note:**

See *Creating Users in Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Manager* for more information about creating a user.

- a. In the left pane, under Administration, click **Users**.
The Search Users page is displayed.
 - b. From the Actions menu, select **Create**. Alternatively, you can click **Create** on the toolbar.
 - c. On the Create User page, enter values for the OIM User fields, and then click **Submit**. A message is displayed stating that the user is created successfully.
3. If you want to provision a target system account to an existing OIM User, then:

 **Note:**

See Searching Users in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Manager* for more information about searching a user.

- a. In the left pane, under Administration, click **Users**.
The Search Users page is displayed.
 - b. Specify a search criteria to search for the OIM User, and then click **Search**.
 - c. From the list of users displayed in the search results, select the OIM User. The user details page is displayed on the right pane.
4. On the Account tab, click **Request Accounts**.
5. In the Catalog page, search for and add to cart the application instance (in other words, the account to be provisioned), and then click **Checkout**.
6. Specify value for fields in the application form and then click **Ready to Submit**.
7. Click **Submit**.
8. If you want to provision entitlements, then:
- a. On the Entitlements tab, click **Request Entitlements**.
 - b. In the Catalog page, search for and add to cart the entitlement, and then click **Checkout**.
 - c. Click **Submit**.

6.7 Extending the Connector for DB2

The following sections describe procedures that you can perform to extend the functionality of the connector for addressing your specific business requirements:

 **Note:**

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

- [Modifying Predefined Queries or Creating New Queries for DB2](#)
- [Configuring Queries to Add Support for Custom Parameters and Lookup Fields for DB2](#)
- [About Configuring the Connector for Multiple Installations of DB2](#)
- [About Configuring the Connector for Multiple Trusted Source Reconciliation from DB2](#)
- [Configuring Validation of Data During Reconciliation and Provisioning for DB2](#)
- [Configuring Transformation of Data During User Reconciliation for DB2](#)
- [Configuring Resource Exclusion Lists for DB2](#)
- [Setting Up Action Scripts for DB2](#)

6.7.1 Modifying Predefined Queries or Creating New Queries for DB2

The following sections discuss the syntax and guidelines that you must apply while modifying the predefined queries or creating new queries:

- [About the Queries for DB2](#)
- [Syntax of Provisioning Queries for DB2](#)
- [Syntax of Reconciliation Queries for DB2](#)
- [Syntax of List of Values Queries for DB2](#)

6.7.1.1 About the Queries for DB2

Predefined queries are provided to reconcile target system user records, synchronize lookup field values with Oracle Identity Manager, and for provisioning operations. You can modify the predefined queries or add your own queries.

The query files are included in a JAR file in the bundle directory of the connector installation media. For example, `bundle/org.identityconnectors.dbum-1.0.1116.jar`.

The connector includes the following types of queries:

- **Provisioning Queries**
They are used for create, update, and delete operations. The query file is `scripts/db2/Provisioning.queries`.
- **List of Values Search Queries**
They are used for reconciliation of lookup definitions. A list of value query operates on a set of values for fields such as profiles, privileges, roles, and tablespaces. The query file is `scripts/db2/LoVSearch.queries`.
- **Account Search Queries**
They are used for full and delete reconciliation operations. An account search query operates on account and group searches with various conditions. The query file is `scripts/db2/Search.queries`.

 **Note:**

The stored procedure OUT parameters cannot be configured for write-back on the process form. The returned values cannot be used for any connector operations.

6.7.1.2 Syntax of Provisioning Queries for DB2

The following is the syntax of the queries used for provisioning operations:

```

QUERYID {
Query="QUERY"
QueryType="QUERYTYPE"
Parameters=["PARAM1":"PARAMDEFN1", "PARAM2":"PARAMDEFN2"...]
ExtensionJoin="EXTENSIONJOIN"
ExtensionSeparator="EXTENSIONSEPARATOR"
QueryExtensions=["EXTENSION1", "EXTENSION2"...]
}

```

For example:

```

CREATE_USER {
  Query="GRANT CONNECT, DBADM, CREATETAB, BINDADD, CREATE_NOT_FENCED_ROUTINE, "+
    "IMPLICIT_SCHEMA, LOAD, CREATE_EXTERNAL_ROUTINE, QUIESCE_CONNECT ON "+
    "DATABASE TO {userType} {__NAME__}"
  QueryType="SQL"
  Parameters=["__NAME__": "Type:String, Direction:IN",
    "userType": "Type:String, Direction:IN, Tags:EXCLUDE_VALIDATION"]
  QueryExtensions=[]
}

```

In this syntax:

- **QUERYID** refers to the unique name of the query.
For example: CREATE_USER
- **QUERY** refers to the main query.
For example: Query="GRANT
CONNECT, DBADM, CREATETAB, BINDADD, CREATE_NOT_FENCED_ROUTINE, "+
"IMPLICIT_SCHEMA, LOAD, CREATE_EXTERNAL_ROUTINE, QUIESCE_CONNECT ON "+
"DATABASE TO {userType} {__NAME__}"
- **QueryType** refers to the type of the main query, either an SQL query or a stored procedure. The value of **QUERYTYPE** can be SQL or StoredProc.
For example: QueryType="SQL"
- **Parameters** refers to the list of comma separated parameters and parameter definitions used with the main query, represented by "PARAM1":"PARAMDEFN1", "PARAM2":"PARAMDEFN2", and so on.

For example: Parameters=["__NAME__": "Type:String,Direction:IN", "userType": "Type:String,Direction:IN,Tags:EXCLUDE_VALIDATION"]

A parameter can have the following attributes:

- Type is the type of the parameter.
- Direction is the flow of data from the query to or from the parameter. It can have a value of IN, OUT, or INOUT.
- TAGS is the enclosure characters that are applied to each parameter before the query is processed. It can have a value of DOUBLEQUOTES, QUOTES, UPPERCASE, LOWERCASE, or EXCLUDE_VALIDATION.

If you want to use multiple tags, you must encapsulate the tags in escaped quotes and separate them by commas. However, you must not use DOUBLEQUOTES with QUOTES or UPPERCASE with LOWERCASE in the same query.

For example: "Type:String,TAGS:\"DOUBLEQUOTES,UPPERCASE\""

- ExtensionJoin (optional) refers to the operator, represented by *EXTENSIONJOIN*, used to join the main query with query extensions.

For example: ExtensionJoin=","

- ExtensionSeparator (optional) refers to the delimiter between query extensions, represented by *EXTENSIONSEPARATOR*.

For example: ExtensionSeparator=","

- QueryExtensions (optional) refers to the extensions that must be appended to the main query, represented by *EXTENSION1*, *EXTENSION2*, and so on.

During a provisioning operation, the connector combines all these components to the following query:

```
QUERY PARAM1, PARAM2... [EXTENSIONJOIN [EXTENSION1 EXTENSIONSEPARATOR
EXTENSION2 EXTENSIONSEPARATOR...]]
```

For example:

```
GRANT CONNECT,DBADM,CREATETAB,BINDADD,CREATE_NOT_FENCED_ROUTINE,"+
"IMPLICIT_SCHEMA,LOAD,CREATE_EXTERNAL_ROUTINE,QUIESCE_CONNECT ON "+ "DATABASE TO
{userType} {__NAME__}
```

Table 6-15 lists the script selection logic of the provisioning queries:

Table 6-15 Script Section Logic for DB2 Provisioning Queries

Operation	Selection Logic	Query IDs
CREATE	CREATE_OBJECTTYPE	CREATE_USER
DELETE	DELETE_OBJECTTYPE	DELETE_USER
ENABLE	ENABLE_OBJECTTYPE	ENABLE_USER
DISABLE	DISABLE_OBJECTTYPE	DISABLE_USER
RESET PASSWORD	SET_PASSWORD	SET_PASSWORD
ADD CHILD VALUES	UPDATE_ADD_ATTRIBUTE	UPDATE_ADD_TABLESPACES UPDATE_ADD_SCHEMAS

Table 6-15 (Cont.) Script Section Logic for DB2 Provisioning Queries

Operation	Selection Logic	Query IDs
REMOVE CHILD VALUES	UPDATE_REVOKE_ATTRIBUTE	UPDATE_REVOKE_TABLESPACES UPDATE_REVOKE_SCHEMAS

6.7.1.3 Syntax of Reconciliation Queries for DB2

The following is the syntax of the search queries used during reconciliation operations:

```

QUERYID {
Query="QUERY"
QueryType="QUERYTYPE"
Parameters=["PARAM1":"PARAMDEFN1", "PARAM2":"PARAMDEFN2"...]
ExtensionJoin="EXTENSIONJOIN"
ExtensionSeparator="EXTENSIONSEPARATOR"
QueryExtensions=["EXTENSION1", "EXTENSION2"...]
}

```

For example:

```

SEARCH_USER {
  Query="SELECT {__UID__}, {userType} FROM SYSIBM.SYSDBAUTH {filter}"
  QueryType="SQL"
  Parameters=["__UID__": "Type:String, Direction:OUT, ColName:GRANTEE",
             "userType": "Type:String, Direction:OUT, ColName:GRANTEETYPE"]

  QueryExtensions=["SEARCH_USER_TABLESPACE", "SEARCH_USER_SCHEMA", "SEARCH_USER_STATU
S"]
}

```

In this syntax:

- *QUERYID* refers to the unique name of the query.

For example: SEARCH_USER

QUERYID can be one of the following values:

- SEARCH_USER
- BATCHED_SEARCH_USER
- SEARCH_USER_SCHEMA
- SEARCH_USER_TABLESPACE
- SEARCH_USER_STATUS

- *QUERY* refers to the main query.

For example: Query="SELECT {__UID__}, {userType} FROM SYSIBM.SYSDBAUTH {filter}"

- `QueryType` refers to the type of the main query, either an SQL query, a stored procedure, or a query extension. The value of `QUERYTYPE` can be `SQL`, `StoredProc`, or `QUERYEXTENSION`.

For example: `QueryType="SQL"`

- `Parameters` refers to the list of comma separated parameters and parameter definitions used with the main query, represented by `"PARAM1":"PARAMDEFN1"`, `"PARAM2":"PARAMDEFN2"`, and so on.

For example:

```
Parameters=["__UID__": "Type:String,Direction:OUT,ColName:GRANTEE",
"userType": "Type:String,Direction:OUT,ColName:GRANTEETYPE"]
```

A parameter can have the following attributes:

- `Type` is the type of the parameter.
- `Direction` is the flow of data from the query to or from the parameter. It can have a value of `IN`, `OUT`, or `INOUT`.
- `ColName` is the column name in the target system corresponding to the parameter in the query.
- `ColQuery` is the query used to fetch values for the corresponding query parameter.
- `ExtensionJoin` (optional) refers to the operator, represented by `EXTENSIONJOIN`, used to join the main query with query extensions.

For example: `ExtensionJoin=", "`

- `ExtensionSeparator` (optional) refers to the delimiter between query extensions, represented by `EXTENSIONSEPARATOR`.

For example: `ExtensionSeparator=", "`

- `QueryExtensions` (optional) refers to the extensions that must be appended to the main query, represented by `EXTENSION1`, `EXTENSION2`, and so on.

For example:

```
QueryExtensions=["SEARCH_USER_TABLESPACE", "SEARCH_USER_SCHEMA", "SEARCH_USER_STATUS"]
```

During a reconciliation operation, the connector combines all these components to the following query:

```
QUERY PARAM1, PARAM2... [EXTENSIONJOIN [EXTENSION1 EXTENSIONSEPARATOR
EXTENSION2 EXTENSIONSEPARATOR...]]
```

For example:

```
SELECT {__UID__}, {userType} FROM SYSIBM.SYSDBAUTH {filter}
SEARCH_USER_TABLESPACE, SEARCH_USER_SCHEMA, SEARCH_USER_STATUS
```

6.7.1.4 Syntax of List of Values Queries for DB2

If a search query is performed on account types, such as User Name, then the query is considered as a reconciliation query. If a search query is performed on any other object, then the query is considered as a list of values query.

The following is the syntax of the list of values queries used for lookup field synchronization:

```
OBJECTTYPE = "QUERY"
```

For example:

```
__TABLESPACES__="SELECT DISTINCT tbspace FROM syscat.tablespaces"
```

In this syntax:

- *OBJECTTYPE* refers to the lookup field attribute.
For example: `__TABLESPACES__` and `__SCHEMAS__`
- *QUERY* refers to the query used for fetching a lookup field attribute.
For example: `SELECT DISTINCT tbspace FROM syscat.tablespaces`

The list of values queries return values that are used as lookup field entries. By default, the connector includes dedicated scheduled job for each lookup definition. To use a custom lookup definition, you must add custom fields in the query file.

6.7.2 Configuring Queries to Add Support for Custom Parameters and Lookup Fields for DB2

The connector uses preconfigured queries for connector operations such as create, delete, and search. You can add custom parameters and lookup definition fields as per your requirements.

The procedure to add a parameter or a lookup definition field to a query file is discussed in the following sections:

- [Updating the Query Files for DB2](#)
- [Configuring Oracle Identity Manager](#)

6.7.2.1 Updating the Query Files for DB2

To update the query files:

1. If the connector is already installed, run the Oracle Identity Manager Download JARs utility to download the connector bundle JAR file from the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

Note:

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

For Microsoft Windows:

```
OIM_HOME/server/bin/DownloadJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/DownloadJars.sh
```

When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Manager administrator, URL of the Oracle Identity Manager host computer, context factory value, type of JAR file being downloaded, and the

location from which the JAR file is to be downloaded. Select ICFBundle as the JAR type.

2. Copy the bundle JAR file in a temporary directory.

Sample JAR file: `bundle/org.identityconnectors.dbum-1.0.1116.jar`

Sample temporary directory: `c:\temp`

3. Run the following command to extract the connector bundle JAR file:

```
jar -xvf org.identityconnectors.dbum-1.0.1116.jar
```

 **Note:**

You can also run the WinZip or WinRAR utility to extract the contents from the JAR file.

4. Delete the bundle JAR file in the temporary directory.
5. Update the value of **ConnectorBundle-Version** in the manifest file, META-INF/MANIFEST.MF, to a new value.

For example:

```
ConnectorBundle-Version: 1.0.1117
```

6. Depending on your requirement, update the query files with new parameters as per the query syntax described in [Modifying Predefined Queries](#) or [Creating New Queries for DB2](#).

For example, if you want to add a new parameter, CUSTOM_ATTRIBUTE, to the CREATE_USER provisioning query:

- a. Open the provisioning query file in a text editor.

Sample query file:

```
c:\temp\bundle\org.identityconnectors.dbum-1.0.1116\scripts\db2\Provisioning.queries
```

- b. Add the parameter, CUSTOM_ATTRIBUTE, to the CREATE_USER query.

The following is a sample updated query:

```
CREATE_USER {
  Query="GRANT
CONNECT, DBADM, CREATETAB, BINDADD, CREATE_NOT_FENCED_ROUTINE, "+
"IMPLICIT_SCHEMA, LOAD, CREATE_EXTERNAL_ROUTINE, QUIESCE_CONNECT ON "+
"DATABASE TO {userType} {__NAME__} {CUSTOM_ATTRIBUTE}"
  QueryType="SQL"
  Parameters=["__NAME__": "Type:String, Direction:IN",
"userType": "Type:String, Direction:IN, Tags:EXCLUDE_VALIDATION",
"CUSTOM_ATTRIBUTE": "Type:String, Direction:IN"]
  QueryExtensions=[]
}
```

- c. Save and close the query file.
7. Create a new bundle JAR file that contains the updated manifest file and the provisioning query file as follows:

- a. Open the command prompt and navigate to the temporary directory:

```
c:\temp
```

- b. Run the following command:

```
jar -cvfm org.identityconnectors.dbum-1.0.1117.jar *
```

The new connector bundle JAR name contains the new bundle version.

8. In the case of a remote connector server, copy the new bundle JAR file in the bundles directory of the remote connector server, instead of posting the JAR file to the Oracle Identity Manager database. Skip to Step 10.
9. Run the Oracle Identity Manager Update JARs utility to update the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

If you have installed both the Oracle and DB2 connectors on the same Oracle Identity Manager, then ensure that all third-party JAR files are part of the `/lib` directory in the connector bundle JAR file.

For Microsoft Windows:

```
OIM_HOME/server/bin/UpdateJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UpdateJars.sh
```

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

10. Update the configuration lookup with the new bundle version.

For example, you can update the `Lookup.DBUM.DB2.Configuration` lookup definition.

6.7.2.2 Configuring Oracle Identity Manager

You can skip this procedure if the parameter you added already exists as a default form field in Oracle Identity Manager.

To configure Oracle Identity Manager for adding a parameter:

1. Log into Oracle Identity Manager Design Console.
2. Create a new version of the process form:
 - a. Expand **Development Tools**.
 - b. Double-click **Form Designer**.
 - c. Search for and open the **UD_DB_DB2_U** process form.
 - d. Click **Create New Version**.

On the Create a new version dialog box, enter a new version in the Label field, and then click the save icon.

3. Add the new field on the process form.
 - a. Click **Add**.

A field is added to the list. Enter the details of the field.

For example, if you are adding the CustomAttribute1 field, enter UD_DB_DB2_U_CUSTOM1 in the **Name** field and then enter the rest of the details of this field.
 - b. Click the save icon and then click **Make Version Active**.
4. If you are using Oracle Identity Manager release 11.1.2.x or later, then all changes made to the Form Designer of the Design Console must be done in a new UI form as follows:
 - a. Log in to Oracle Identity System Administration.
 - b. Create and active a sandbox.
 - c. Create a new UI form to view the newly added field along with the rest of the fields. See *Creating Forms By Using the Form Designer in Oracle Fusion Middleware Administering Oracle Identity Manager*.
 - d. Associate the newly created UI form with the application instance of your target system. To do so, open the existing application instance for your resource, from the Form field, select the form (created in Step 4.c), and then save the application instance.
 - e. Publish the sandbox as described in *Publishing a Sandbox of Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.
5. Create an entry for the field in the lookup definition for provisioning as follows:
 - a. Expand **Administration**.
 - b. Double-click **Lookup Definition**.
 - c. Search for and open the **Lookup.DBUM.DB2.UM.ProvAttrMap** lookup definition.
 - d. Click **Add** and enter the Code Key and Decode values for the field.

The Code Key value must be the form field name. The Decode value must be the attribute name on the target system.

For example, enter `Custom Attribute 1` in the **Code Key** field and then enter `CustomAttribute1` in the **Decode** field.
 - e. Click the save icon.
6. Create a process task to update the new field Custom Attribute 1 as follows:
 - a. Expand **Process Management**.
 - b. Double-click **Process Definition** and open the **DB2 DB User** process definition.
 - c. Click **Add** and enter the task name, for example, `Custom Attribute 1 Updated`, and the task description.
 - d. In the Task Properties section, select Conditional and Allow Multiple Instances fields and click the save icon.
 - e. On the Integration tab, click **Add**, and then click **Adapter**.
 - f. Select the **adpDB2UPDATECHILDTABLEVALUES** adapter, click the save icon, and then click **OK** in the message that is displayed.

- g. To map the adapter variables listed in this table, select the adapter, click **Map**, and then specify the data given in the following table:

Variable Name	Data Type	Map To	Qualifier	Literal Value
Adapter return value	Object	Response code	NA	NA
attributeName	String	Literal	String	Custom Attribute 1
itRes	String	Literal	String	UD_DB_DB2_U_ITRES
objectType	String	Literal	String	User
processInstanceKey	Long	Process Data	Process Instance	NA

- h. On the Responses tab, click **Add** to add the following response codes:

Code Name	Description	Status
ERROR	Error occurred	R
UNKNOWN	An unknown response was received	R
SUCCESS	Operation completed	C

- i. Click the save icon and then close the dialog box.

6.7.3 About Configuring the Connector for Multiple Installations of DB2

You might want to configure the connector for multiple installations of the target system. The following example illustrates this requirement:

The London and New York offices of Example Multinational Inc. have their own installations of the target system. The company has recently installed Oracle Identity Manager, and they want to configure Oracle Identity Manager to link all the installations of the target system.

To meet the requirement posed by such a scenario, you can create copies of connector objects, such as the IT resource and resource object.

The decision to create a copy of a connector object might be based on a requirement. For example, an IT resource can hold connection information for one target system installation. Therefore, it is mandatory to create a copy of the IT resource for each target system installation.

With some other connector objects, you do not need to create copies at all. For example, a single attribute-mapping lookup definition can be used for all installations of the target system.

All connector objects are linked. For example, a scheduled job holds the name of the IT resource. Similarly, the IT resource for a target system such as DB2 holds the name of the configuration lookup definition, Lookup.DBUM.DB2.Configuration. If you create a copy of an object, then you must specify the name of the copy in associated connector objects.

 **Note:**

- To reconcile data from a particular target system installation, specify the name of the IT resource for that target system installation as the value of the scheduled job attribute that holds the IT resource name. For example, you enter the name of the IT resource as the value of the IT resource attribute of the scheduled job that you run.
- When you use Identity Self Service to perform provisioning, you can specify the IT resource corresponding to the target system installation to which you want to provision the user.

Table 6-16 lists associations between connector objects whose copies can be created and the other objects that reference these objects. When you create a copy of a connector object, use this information to change the associations of that object with other objects.

 **Note:**

- On a particular Oracle Identity Manager installation, if you create a copy of a connector object, then you must set a unique name for it.
- If you are using Oracle Identity Manager release 11.1.2.x or later, then in addition to the procedure described in this section, you must create an application instance for each IT resource. See [Configuring Oracle Identity Manager Release 11.1.2 or Later](#) for information on creating an application instance.

Table 6-16 Connector Objects and Their Associations

Connector Object	Name	Referenced By	Comments on Creating a Copy
IT resource	DB2 DB	<ul style="list-style-type: none"> • UD_DB_DB2_U (process form) • Scheduled tasks 	Create a copy of the IT resource with a different name.
Resource object	DB2 DB User DB2 DB Trusted	All connector operations	<p>It is optional to create a copy of the resource object. If you are reconciling the same set of attributes from all installations of the target system, then you need not create a copy of the resource object.</p> <p>Note: Create copies of the resource object only if there are differences in attributes between the various installations of the target system.</p>
Scheduled Jobs	There are many scheduled jobs for different purposes.	NA	You can use the scheduled jobs with the same names. However, you must update the values of the parameters depending on the target system you want to use.

Table 6-16 (Cont.) Connector Objects and Their Associations

Connector Object	Name	Referenced By	Comments on Creating a Copy
Process definition	DB2 DB User	NA	<p>It is optional to create a copy of the process definition. If you are reconciling or provisioning the same set of attributes from all installations of the target system, then you need not create a copy of the process definition.</p> <p>Note: Create copies of the process form only if there are differences in attributes between the various installations of the target system.</p>
Process form	UD_DB_DB2_U	DB2 DB User (Process definition)	<p>It is optional to create a copy of the process form. If you are provisioning the same set of attributes from all installations of the target system, then you need not create a copy of the process definition.</p> <p>Note: Create copies of the process form only if there are differences in attributes between the various installations of the target system.</p>
Child process form	<ul style="list-style-type: none"> • UD_DB_DB2_S • UD_DB_DB2_T 	<ul style="list-style-type: none"> • DB2 DB User (Process definition) • UD_DB_DB2_U (Process form) 	<p>It is optional to create a copy of the child process form. If you are provisioning a new set of child data, then you need to create a copy of the child and parent process forms. Then, assign the newly created child process form to the newly created parent process form.</p>
Configuration lookup definition for a target system configured as a target resource	Lookup.DBUM.D B2.Configuration	DB2 (IT resource)	<p>It is optional to create a copy of the configuration lookup definition. If you are provisioning and reconciling the same set of attributes in all installations of the target system (configured as a target resource), then you need not create a copy of the configuration lookup definition.</p> <p>Note: Create copies of the configuration lookup definition only if there are differences in attributes between the various installations of the target system and you have created a new process form.</p>

Table 6-16 (Cont.) Connector Objects and Their Associations

Connector Object	Name	Referenced By	Comments on Creating a Copy
Configuration lookup definition for a target system configured as a trusted source	Lookup.DBUM.D B2.Configuration. Trusted	DB2 (IT resource)	<p>It is optional to create a copy of the configuration lookup definition. If you are reconciling the same set of attributes in all installations of the target system (configured as a trusted source), then you need not create a copy of the configuration lookup definition.</p> <p>Note: Create copies of the configuration lookup definition for trusted source only if there are differences in attributes between the various installations of the target system and you have created a new process form.</p>
Resource object attributes mapping lookup definition (for target resource)	Lookup.DBUM.D B2.UM.ReconAttr Map	NA	<p>It is optional to create a copy of resource object attribute mapping lookup definition. If you are reconciling the same set of attributes in all installations of the target system, then you need not to create a copy of resource object attribute mapping lookup.</p> <p>Note: Create copies of this lookup definition only if there are differences in attributes between the two installations of the target system.</p>
Configuration lookup definition for a target system configured as a trusted source	Lookup.DBUM.D B2.UM.ReconAttr Map.Trusted	DB2 (IT resource)	<p>It is optional to create a copy of the configuration lookup definition. If you are reconciling the same set of attributes in all installations of the target system (configured as a trusted source), then you need not create a copy of the configuration lookup definition.</p> <p>Note: Create copies of the configuration lookup definition for trusted source only if there are differences in attributes between the various installations of the target system and you have created a new process form.</p>

6.7.4 About Configuring the Connector for Multiple Trusted Source Reconciliation from DB2

 **Note:**

This connector supports multiple trusted source reconciliation.

This section describes an optional procedure. Perform this procedure only if you want to configure the connector for multiple trusted source reconciliation.

The following are examples of scenarios in which there is more than one trusted source for user data in an organization:

- One of the target systems is a trusted source for data about users. The second target system is a trusted source for data about contractors. The third target system is a trusted source for data about interns.
- One target system holds the data of some of the identity fields that constitute an OIM User. Two other systems hold data for the remaining identity fields. In other words, to create an OIM User, data from all three systems would need to be reconciled.

If the operating environment of your organization is similar to that described in either one of these scenarios, then this connector enables you to use the target system as one of the trusted sources of person data in your organization.

See *Managing Reconciliation in Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed information about multiple trusted source reconciliation.

6.7.5 Configuring Validation of Data During Reconciliation and Provisioning for DB2

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

To configure validation of data:

1. Write code that implements the required validation logic in a Java class with a fully qualified domain name (FQDN), such as `org.identityconnectors.dbum.extension.DBUMValidator`.

This validation class must implement the `validate` method. The following sample validation class checks if the value in the First Name attribute contains the number sign (#):

```
package com.validationexample;

import java.util.HashMap;

public class MyValidator {
```

```

    public boolean validate(HashMap hmUserDetails, HashMap hmEntitlementDetails,
String sField) throws ConnectorException {

        /* You must write code to validate attributes. Parent
        * data values can be fetched by using hmUserDetails.get(field)
        * For child data values, loop through the
        * ArrayList/Vector fetched by hmEntitlementDetails.get("Child
Table")
        * Depending on the outcome of the validation operation,
        * the code must return true or false.
        */

        /*
        * In this sample code, the value "false" is returned if the field
        * contains the number sign (#). Otherwise, the value "true" is
        * returned.
        */
        boolean valid = true;
        String sFirstName = (String) hmUserDetails.get(sField);
        for (int i = 0; i < sFirstName.length(); i++) {
            if (sFirstName.charAt(i) == '#') {
                valid = false;
                break;
            }
        }
        return valid;
    }
}

```

2. Log in to the Design Console.
3. Search for and open one of the lookup definitions (or create a new lookup) listed in [Lookup Definition for Validation of Data in DB2](#).

For example, **Lookup.DBUM.DB2.UM.ProvValidations**.

 **Note:**

If you cannot find these lookup definitions, create new lookup definitions.

4. In the **Code Key** column, enter the resource object field name that you want to validate. For example, Username.
5. In the **Decode** column, enter the class name. For example, org.identityconnectors.dbum.extension.DBUMValidator.
6. Save the changes to the lookup definition.
7. Search for and open the configuration lookup definition for the target system you use. For example, **Lookup.DBUM.DB2.UM.Configuration**.
8. In the **Code Key** column, enter one of the following entries:
 - To configure validation of data for reconciliation:
Recon Validation Lookup
 - To configure validation of data for provisioning:
Provisioning Validation Lookup

- In the **Decode** column, enter the name of the lookup you updated or created in step 3.
For example, **Lookup.DBUM.DB2.UM.ProvValidations**.
- Save the changes to the lookup definition.
- Create a JAR with the class and upload it to the Oracle Identity Manager database as follows:

Run the Oracle Identity Manager Upload JARs utility to post the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

```
OIM_HOME/server/bin/UploadJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UploadJars.sh
```

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

- Run the PurgeCache utility to clear content related to request datasets from the server cache.
- Perform reconciliation or provisioning to verify validation for the field, for example, Username.

6.7.6 Configuring Transformation of Data During User Reconciliation for DB2

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

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

- Write code that implements the required transformation logic in a Java class with a fully qualified domain name (FQDN), such as
`org.identityconnectors.dbum.extension.DBUMTransformation`.

This transformation class must implement the transform method. The following sample transformation class modifies the Username attribute by using values fetched from the `__NAME__` attribute of the target system:

```
package com.transformationexample;  
  
import java.util.HashMap;
```

```

public class MyTransformer {
    public Object transform(HashMap hmUserDetails, HashMap hmEntitlementDetails,
String sField) throws ConnectorException {
        /*
        * You must write code to transform the attributes.
        * Parent data attribute values can be fetched by
        * using hmUserDetails.get("Field Name").
        * To fetch child data values, loop through the
        * ArrayList/Vector fetched by hmEntitlementDetails.get("Child
Table")
        * Return the transformed attribute.
        */
        String sUserName = (String) hmUserDetails.get("__NAME__");
        return sUserName + "@example.com";
    }
}

```

2. Log in to the Design Console.
3. Search for and open one of the lookup definitions (or create a new lookup) listed in [Lookup Definitions for Transformation of Data in DB2](#).

For example, **Lookup.DBUM.DB2.UM.ReconTransformations**.

 **Note:**

If you cannot find these lookup definitions, create new lookup definitions.

4. In the **Code Key** column, enter the resource object field name you want to transform. For example, Username.
5. In the **Decode** column, enter the class name. For example, org.identityconnectors.dbum.extension.DBUMTransformation.
6. Save the changes to the lookup definition.
7. Search for and open the **Lookup.DBUM.DB2.UM.Configuration** lookup definition.
8. In the **Code Key** column, enter Recon Transformation Lookup.
9. In the **Decode** column, enter the name of the lookup you updated or created in step 3. For example, **Lookup.DBUM.DB2.UM.ReconTransformations**. For trusted mode, use **Lookup.DBUM.DB2.UM.ReconTransformations.Trusted**.
10. Save the changes to the lookup definition.
11. Create a JAR with the class and upload it to the Oracle Identity Manager database as follows:

Run the Oracle Identity Manager Upload JARs utility to post the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

```
OIM_HOME/server/bin/UploadJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UploadJars.sh
```

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

12. Run the PurgeCache utility to clear content related to request datasets from the server cache.
13. Perform reconciliation to verify transformation of the field, for example, SimpleDisplayName.

6.7.7 Configuring Resource Exclusion Lists for DB2

You can specify a list of accounts that must be excluded from reconciliation and provisioning operations. Accounts whose user IDs you specify in the exclusion list are not affected by reconciliation and provisioning operations.

In one of the lookup definitions for exclusion lists, enter the user IDs of target system accounts for which you do not want to perform provisioning and reconciliation operations. See [Lookup Definitions for Exclusion Lists for DB2](#) for information about the lookup definitions and the format of the entries in these lookups.

To add entries in the lookup for exclusions during provisioning and reconciliation operations for DB2:

1. On the Design Console, expand **Administration** and then double-click **Lookup Definition**.
2. Search for and open the **Lookup.DBUM.DB2.UM.ExclusionList** lookup definition.
3. Click **Add**.
4. In the Code Key column, enter the resource object field name on which the exclusion list is applied. In the Decode column, enter the corresponding ID of the record to exclude.

For example, if you do not want to provision users with the user ID User001, then you must populate the lookup definition with the following values:

Code Key	Decode
User Name	User001

 **Note:**

If you want to specify a list of accounts that must be excluded during reconciliation or provisioning, the code key value being specified here must be exactly as the corresponding code key value in the Lookup.DBUM.DB2.UM.ReconAttrMap lookup definition, or in the Lookup.DBUM.DB2.UM.ProvAttrMap lookup definition, respectively.

5. If there is more than one user ID to exclude, then in the decode column, enter a list of all user IDs to exclude. Note that each User ID must be separated by a vertical bar (|).

For example, if you do not want to provision users with user IDs User001, User002, and User088 then you must populate the lookup definition with the following values:

Code Key	Decode
User Name	User001 User002 User088

You can also perform pattern matching to exclude user accounts. You can specify regular expressions supported by the representation in the `java.util.regex.Pattern` class.

 **See Also:**

For information about the supported patterns, visit <http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html>

For example, if you do not want to provision users matching any of the user IDs User001, User002, and User088, then you must populate the lookup definition with the following values:

Code Key	Decode
User Name[PATTERN]	User001 User002 User088

If you do not want to provision users whose user IDs start with 00012, then you must populate the lookup definition with the following values:

Code Key	Decode
User Name[PATTERN]	00012*

6. Click the save icon.

6.7.8 Setting Up Action Scripts for DB2

Learn about action scripts and how to configure them to run before or after the create, update, or delete an account provisioning operations.

This section provides information about the following topics:

- [About Action Scripts for DB2](#)

- [Configuring Action Scripts for DB2](#)

6.7.8.1 About Action Scripts for DB2

Action scripts are scripts that you can configure to run before or after the create, update, or delete an account provisioning operations. For example, you could configure a script to run before every user creation. In another scenario, suppose you have a table called `AUDIT_USERLOG` where you want to log user creation activities performed only by the connector. Then, you could create and use after create script for adding data to this table after create operation.

Note:

To configure a before or after action, your connector must support running scripts. An exception is Groovy (with target set to **Connector**), which the Identity Connector Framework (ICF) supports by default for all converged connectors.

Every connector should specify which scripting language and which target it supports. This connector supports the following script:

- **shell:** shell script
- **target:** Connector

The target refers to the location where the script is executed. In this case, the script is executed on the same computer (JVM or .NET Runtime) where the connector is deployed. For example, if you deploy the connector on the connector server, the script will be executed on that computer.

That is, if you are using a local framework, the script runs in your JVM. If you are connected to a remote framework, the script runs in the remote JVM or .NET Runtime.

6.7.8.2 Configuring Action Scripts for DB2

To configure the action:

1. Log in to the Design Console.
2. Search for and open the **Lookup.DBUM.DB2.UM.Configuration** lookup definition.
3. Add the following new values:
 - **Code Key:** `Before Create Action Language`
 - **Decode:** Enter the scripting language of the script you want to execute
 - Sample values: `SQL` or `STOREDPROC`
4. Add these new values:
 - **Code Key:** `Before Create Action File`
 - **Decode:** Enter the full path to the file containing the script to be executed (Oracle Identity Manager must be able to access this file.)
 - **Example:** `/home/scripts/testscript.sql`

This script may have a query as follows:

```
INSERT INTO AUDIT_USERLOG VALUES (['__NAME__'], CURRENT_TIMESTAMP))
```

5. Add these new values:

- **Code Key:** Before Create Action Target
- **Decode:** Connector

6. Save the lookup definition.

Now, this action will be executed every time you create a user. You must configure these three values for each action you want to execute.

7

Using and Extending the Connector for Sybase

You can use the Database User Management Connector for Sybase for performing reconciliation and provisioning operations after configuring it to meet your requirements. You can also extend the functionality of the connector to address your specific business requirements.

This chapter contains the following topics:



Note:

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

For Oracle Identity Manager hosted on a Microsoft Windows computer, if you have a previously installed connector, then you must extract the connector bundle zip file again before installing a new connector.

- [Configuring Secure Communication Between Sybase and Oracle Identity Manager](#)
- [Guidelines on Determining Values for the JDBC URL and Connection Properties Parameters for Sybase](#)
- [Lookup Definitions for Sybase](#)
- [Scheduled Jobs for Sybase](#)
- [Reconciliation from Sybase](#)
- [Provisioning for Sybase](#)
- [Extending the Connector for Sybase](#)

7.1 Configuring Secure Communication Between Sybase and Oracle Identity Manager



Note:

It is recommended that you perform the procedure described in this section to secure communication between the target system and Oracle Identity Manager.

Perform the following steps to configure secure communication between Sybase and Oracle Identity Manager:

1. See Sybase Adaptive Server Enterprise documentation for information about enabling SSL communication between Sybase and a client system. In this context, the client is Oracle Identity Manager.

Export the certificate on the Sybase host computer.

2. Copy the certificate to the Oracle Identity Manager host computer.
3. Import the certificate into the JVM truststore of the application server on which Oracle Identity Manager is running.

To import the certificate into the truststore, run the following command:

```
..\..\bin\keytool -import -file FILE_LOCATION -keystore TRUSTSTORE_LOCATION -storepass TRUSTSTORE_PASSWORD -trustcacerts -alias ALIAS
```

In this command:

- Replace *FILE_LOCATION* with the full path and name of the certificate file.
- Replace *ALIAS* with an alias for the certificate.
- Replace *TRUSTSTORE_PASSWORD* with a password for the truststore.
- Replace *TRUSTSTORE_LOCATION* with one of the truststore paths from [Table 7-1](#). This table shows the location of the truststore for each of the supported application servers.

Table 7-1 Truststore Locations on Supported Application Servers

Application Server	Truststore Location
Oracle WebLogic Server	<ul style="list-style-type: none"> • If you are using Oracle jrockit_R27.3.1-jdk, then import the certificate into the keystore in the following directory: <i>JROCKIT_HOME</i>/jre/lib/security • If you are using the default Oracle WebLogic Server JDK, then import the certificate into the keystore in following directory: <i>WEBLOGIC_HOME</i>/java/jre/lib/security/cacerts • If you are using a JDK other than Oracle jrockit_R27.3.1-jdk or Oracle WebLogic Server JDK, then import the certificate into your keystore at the following directory: <i>JAVA_HOME</i>/jre/lib/security/cacerts

4. To enable secure communication between Sybase and Oracle Identity Manager, set the value of the UseSSL IT resource parameter to `true`. You must provide a value for this parameter while performing the procedure described in [Configuring the IT Resource for the Connector Server](#).

7.2 Guidelines on Determining Values for the JDBC URL and Connection Properties Parameters for Sybase

This section discusses the JDBC URL and Connection Properties parameters. You apply the information in this section while performing the procedure described in [Configuring the IT Resource for the Target System](#).

The following are guidelines on specifying the JDBC URL and Connection Properties parameters:

- **JDBC URL parameter**

Enter the following component of the connection URL as the value of the JDBC URL provider:

```
jdbc:sybase:Tds:SERVER_NAME:PORT_NUMBER/DATABASE_NAME
```

In this format:

- *SERVER_NAME* is the IP address (not the host name) of the target system host computer.
- *PORT_NUMBER* is the port at which the target system database is listening.
- *DATABASE_NAME* is the name of the target system database.

The following is a sample value for the JDBC URL parameter:

```
jdbc:sybase:Tds:172.21.109.62:9050/master
```

- **Connection Properties parameter**

Enter the following component of the connection URL as the value of the Connection Properties parameter:

```
[, PROPERTY=VALUE[, PROPERTY=VALUE]] . . .
```

In this format:

- *PROPERTY* is the name of one or more database connection properties, such as *applicationName* and *disableStatementPooling*.
- *VALUE* is the value of each database connection property whose name you specify by using the *PROPERTY* placeholder.

The following is a sample value for the Connection Properties parameter:

```
databaseName=sales#port=9000
```

7.3 Lookup Definitions for Sybase

This section contains the following information:

- [Lookup Definitions Synchronized with Sybase](#)
- [Lookup Definitions for Configurations for Sybase](#)
- [Lookup Definitions for Attribute Mappings for Sybase Login Entity](#)
- [Lookup Definitions for Attribute Mappings for Sybase User Entity](#)
- [Lookup Definition for Validation of Data in Sybase](#)
- [Lookup Definitions for Transformation of Data in Sybase](#)
- [Lookup Definitions for Exclusion Lists for Sybase](#)

7.3.1 Lookup Definitions Synchronized with Sybase

During a provisioning operation, you use a lookup field on the process form to specify a single value from a set of values. For example, you use the Privileges lookup field to select a privilege to be assigned to a schema from the list of available privileges. When you deploy the connector, lookup definitions corresponding to the lookup fields on the target system are created in Oracle Identity Manager. Lookup field synchronization involves copying additions or changes made to the target system lookup fields into the lookup definitions in Oracle Identity Manager.

The connector provides predefined SQL queries for fetching values from the target system lookup fields into the lookup definitions in Oracle Identity Manager. These predefined SQL queries are stored in the `LoVSearch.queries` file with in the connector bundle.

After lookup definition synchronization, data is stored in the following format:

- Code Key value: `IT_RESOURCE_KEY~LOOKUP_FIELD_ID`

In this format:

- `IT_RESOURCE_KEY` is the numeric code assigned to each IT resource in Oracle Identity Manager.
- `LOOKUP_FIELD_ID` is the target system code assigned to each lookup field entry.

Sample value: `1~SYS_ADM`

- Decode value: `IT_RESOURCE_NAME~LOOKUP_FIELD_ID`

In this format:

- `IT_RESOURCE_NAME` is the name of the IT resource in Oracle Identity Manager.
- `LOOKUP_FIELD_ID` is the target system code assigned to each lookup field entry.

Sample value: `Sybase DB~SYS_ADM`

While performing a provisioning operation in Identity Self Service, you select the IT resource for the target system on which you want to perform the operation. When you perform this action, the lookup definitions on the page are automatically populated with values corresponding to the IT resource (target system installation) that you select. If your environment has multiple installations of the target system, then values corresponding to all IT resources are displayed.

[Table 7-2](#) lists column names of the tables in Sybase that are synchronized with their corresponding lookup definitions in Oracle Identity Manager.

Table 7-2 Lookup Definitions Synchronized with Sybase

Lookup Definition	Target Column Name
Lookup.DBUM.Sybase.Databases	Database Name
Lookup.DBUM.Sybase.DBGroups	Database Group
Lookup.DBUM.Sybase.DefaultLang	Default Language
Lookup.DBUM.Sybase.Roles	Role

7.3.2 Lookup Definitions for Configurations for Sybase

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

This section provides information about the following lookup definitions

- [Lookup.DBUM.Sybase.Configuration](#)

- [Lookup.DBUM.Sybase.Login.Configuration](#)
- [Lookup.DBUM.Sybase.UM.Configuration](#)
- [Lookup.DBUM.Sybase.Configuration.Trusted](#)
- [Lookup.DBUM.Sybase.UM.Configuration.Trusted](#)

7.3.2.1 Lookup.DBUM.Sybase.Configuration

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

Table 7-3 Entries in Lookup.DBUM.Sybase.Configuration

Code Key	Decode Key	Description
Bundle Name	org.identityconnectors.dbum	Name of the connector bundle package Do not modify this entry.
Bundle Version	1.0.1116	Version of the connector bundle class Do not modify this entry.
Connector Name	org.identityconnectors.dbum.DBUMConnector	Name of the connector class Do not modify this entry.
disableValuesSet	"YES"	This entry is used internally. Do not modify this entry.
USERLOGIN Configuration Lookup	Lookup.DBUM.Sybase.Login.Configuration	Name of the lookup definition that contains configuration properties for login entities Do not modify this entry.
User Configuration Lookup	Lookup.DBUM.Sybase.UM.Configuration	Name of the lookup definition that contains user-specific configuration properties Do not modify this entry.

7.3.2.2 Lookup.DBUM.Sybase.Login.Configuration

The Lookup.DBUM.Sybase.Login.Configuration lookup definition holds connector configuration entries for login entities that are used during target resource reconciliation and provisioning operations.

Table 7-4 Entries in Lookup.DBUM.Sybase.Login.Configuration

Code Key	Decode Key
Provisioning Attribute Map	Lookup.DBUM.Sybase.Login.ProvAttrMap
Provisioning Validation Lookup	Lookup.DBUM.Sybase.Login.ProvValidations
Recon Attribute Defaults	Lookup.DBUM.Sybase.Login.ReconDefaults
Recon Attribute Map	Lookup.DBUM.Sybase.Login.ReconAttrMap
Recon Transformation Lookup	Lookup.DBUM.Sybase.Login.ReconTransformations

7.3.2.3 Lookup.DBUM.Sybase.UM.Configuration

The Lookup.DBUM.Sybase.UM.Configuration lookup definition holds user-specific connector configuration entries that are used during target resource reconciliation and provisioning operations.

Table 7-5 Entries in Lookup.DBUM.Sybase.UM.Configuration

Code Key	Decode Key
Provisioning Attribute Map	Lookup.DBUM.Sybase.UM.ProvAttrMap
Provisioning Validation Lookup	Lookup.DBUM.Sybase.UM.ProvValidations
Recon Attribute Defaults	Lookup.DBUM.Sybase.UM.ReconDefaults
Recon Attribute Map	Lookup.DBUM.Sybase.UM.ReconAttrMap
Recon Transformation Lookup	Lookup.DBUM.Sybase.UM.ReconTransformations

7.3.2.4 Lookup.DBUM.Sybase.Configuration.Trusted

The Lookup.DBUM.Sybase.Configuration.Trusted lookup definition holds connector configuration entries that are used during reconciliation and provisioning operations in trusted source mode.



Note:

You cannot add entries to this lookup definition. However, you can modify the Decode values of the existing entries.

Table 7-6 lists the default entries in this lookup definition.

Table 7-6 Entries in the Lookup.DBUM.Sybase.Configuration.Trusted Lookup Definition

Code Key	Decode	Description
Bundle Name	org.identityconnectors.dbum	This entry holds the name of the connector bundle package. Do not modify this entry.
Bundle Version	1.0.1116	This entry holds the version of the connector bundle class. Do not modify this entry.
Connector Name	org.identityconnectors.dbum.DBUMConnector	This entry holds the name of the connector class. Do not modify this entry.
disableValuesSet	"YES"	This entry is used internally. Do not modify this entry.
USERLOGIN Configuration Lookup	Lookup.DBUM.Sybase.UM.Configuration.Trusted	This entry holds the name of the lookup definition that contains login-specific configuration properties. Do not modify this entry.

7.3.2.5 Lookup.DBUM.Sybase.UM.Configuration.Trusted

The Lookup.DBUM.Sybase.UM.Configuration.Trusted lookup definition holds connector configuration entries for Sybase user entity that are used during reconciliation and provisioning operations in trusted source mode.

[Table 7-7](#) lists the default entries in this lookup definition.

Table 7-7 Entries in Lookup.DBUM.Sybase.UM.Configuration.Trusted

Code Key	Dcode Key
Recon Attribute Defaults	Lookup.DBUM.Sybase.UM.ReconDefaults.Trusted
Recon Attribute Map	Lookup.DBUM.Sybase.UM.ReconAttrMap.Trusted
Recon Exclusion List	Lookup.DBUM.Sybase.UM.ExclusionList.Trusted
Recon Transformation Lookup	Lookup.DBUM.Sybase.UM.ReconTransformations.Trusted
Recon Validation Lookup	Lookup.DBUM.Sybase.UM.ReconValidation.Trusted

7.3.3 Lookup Definitions for Attribute Mappings for Sybase Login Entity

This section describes the following lookup definitions:

- [Lookup.DBUM.Sybase.Login.ProvAttrMap](#)
- [Lookup.DBUM.Sybase.Login.ReconAttrMap](#)
- [Lookup.DBUM.Sybase.Login.ReconDefaults](#)

7.3.3.1 Lookup.DBUM.Sybase.Login.ProvAttrMap

The Lookup.DBUM.Sybase.Login.ProvAttrMap lookup definition holds mappings for login entities between process form fields (Code Key values) and target system attributes (Decode values) used during provisioning operations.

[Table 7-8](#) lists the default entries in this lookup definition.

Table 7-8 Entries in Lookup.DBUM.Sybase.Login.ProvAttrMap

Code Key	Decode Key
Default Database[LOOKUP]	defaultDatabase
Default Language[LOOKUP]	defaultLanguage
Full Name	fullName
Password	__PASSWORD__
Login Name	__NAME__
Return ID	__UID__
UD_DB_SYB_R~Role[LOOKUP]	roleList~role~__NAME__

7.3.3.2 Lookup.DBUM.Sybase.Login.ReconAttrMap

The Lookup.DBUM.Sybase.Login.ReconAttrMap lookup definition holds mappings for Sybase login entity between process form fields (Code Key values) and target system attributes (Decode values) used during reconciliation operations.

[Table 7-9](#) lists the default entries in this lookup definition.

Table 7-9 Entries in Lookup.DBUM.Sybase.Login.ReconAttrMap

Code Key	Decode Key
Default Database Name[LOOKUP]	defaultDatabase
Default Language[LOOKUP]	defaultLanguage
Full Name	fullName
Login Name	__UID__
Reference ID	__UID__
Roles List~Role Name[LOOKUP]	roleList
Status	__ENABLE__

7.3.3.3 Lookup.DBUM.Sybase.Login.ReconDefaults

This lookup definition contains the default values for the Oracle Identity Manager user login attributes. You can change these values as per your requirements.

[Table 7-10](#) lists the default entry in this lookup definition.

Table 7-10 Entries in Lookup.DBUM.Sybase.Login.ReconDefaults

Code Key	Decode Key
Status	Enabled

7.3.4 Lookup Definitions for Attribute Mappings for Sybase User Entity

This section describes the following lookup definitions:

- [Lookup.DBUM.Sybase.UM.ProvAttrMap](#)
- [Lookup.DBUM.Sybase.UM.ReconAttrMap](#)
- [Lookup.DBUM.Sybase.UM.ReconDefaults](#)
- [Lookup.DBUM.Sybase.UM.ReconDefaults.Trusted](#)
- [Lookup.DBUM.Sybase.UM.ReconAttrMap.Trusted](#)

7.3.4.1 Lookup.DBUM.Sybase.UM.ProvAttrMap

The Lookup.DBUM.Sybase.UM.ProvAttrMap lookup definition holds user-specific mappings between process form fields (Code Key values) and target system attributes (Decode values) used during provisioning operations.

[Table 7-11](#) lists the default entries in this lookup definition.

Table 7-11 Entries in Lookup.DBUM.Sybase.UM.ProvAttrMap

Code Key	Decode Key
Database Group[LOOKUP]	databaseGroup
Database Name	databaseName
Login Name	loginName
Return ID	__UID__
Username	__NAME__

7.3.4.2 Lookup.DBUM.Sybase.UM.ReconAttrMap

The Lookup.DBUM.Sybase.UM.ReconAttrMap lookup definition holds mappings between resource object fields (Code Key values) and target system attributes (Decode values) used during reconciliation operations.

[Table 7-12](#) lists the default entries in this lookup definition.

Table 7-12 Entries in Lookup.DBUM.Sybase.UM.ReconAttrMap

Code Key	Decode Key
Database Group	databaseGroup
Login Name	loginName
Reference ID	__UID__
User Name	__UID__

7.3.4.3 Lookup.DBUM.Sybase.UM.ReconDefaults

The Lookup.DBUM.Sybase.UM.ReconDefaults lookup definition contains the default values for the Oracle Identity Manager user attributes. You can change these values as per your requirements.:

[Table 7-13](#) lists the default entry in this lookup definition.

Table 7-13 Entries in Lookup.DBUM.Sybase.UM.ReconDefaults

Code Key	Decode Key
Status	Enabled

7.3.4.4 Lookup.DBUM.Sybase.UM.ReconDefaults.Trusted

The Lookup.DBUM.Sybase.UM.ReconDefaults.Trusted lookup definition holds the following entries:

Table 7-14 Entries in Lookup.DBUM.Sybase.UM.ReconDefaults.Trusted

Code Key	Decode Key
Empl Type	Full-Time
Organization Name	Xellerate Users
Status	Active
User Type	End-User

7.3.4.5 Lookup.DBUM.Sybase.UM.ReconAttrMap.Trusted

The Lookup.DBUM.Sybase.UM.ReconAttrMap.Trusted lookup definition holds mappings between resource object fields (Code Key values) and target system attributes (Decode values) used during reconciliation operations in trusted source mode.

[Table 7-15](#) lists the default entries in this lookup definition.

Table 7-15 Entries in Lookup.DBUM.Sybase.UM.ReconAttrMap.Trusted

Code Key	Decode Key
Display Name	fullName
First Name	__UID__
Status[TRUSTED]	__ENABLE__
User ID	__UID__

7.3.5 Lookup Definition for Validation of Data in Sybase

You can use the Lookup.DBUM.Sybase.Login.ProvValidations, Lookup.DBUM.Sybase.UM.ProvValidations, and Lookup.DBUM.Sybase.UM.ReconValidation.Trusted lookup definitions to configure validation of data during provisioning and reconciliation operations.

[Configuring Validation of Data During Reconciliation and Provisioning for Sybase](#) describes the procedure to add entries in these lookup definitions.

7.3.6 Lookup Definitions for Transformation of Data in Sybase

Depending on how the target system is configured, you can use one of the following lookups to enable transformation of data during reconciliation operations:

- For Sybase login entity: Lookup.DBUM.Sybase.Login.ReconTransformations

Table 7-16 Entries in Lookup.DBUM.Sybase.Login.ReconTransformations

Code Key	Decode Key
Roles List	oracle.iam.connectors.dbum.transformations.SybaseRoleTransformation

- For Sybase user entity: Lookup.DBUM.Sybase.UM.ReconTransformations
- For Sybase user entity in trusted source mode:
Lookup.DBUM.Sybase.UM.ReconTransformations.Trusted

[Configuring Transformation of Data During User Reconciliation for Sybase](#) describes the procedure to add entries in these lookup definitions.

7.3.7 Lookup Definitions for Exclusion Lists for Sybase

There are no OOTB lookup definitions provided for exclusion lists for Sybase, apart from Lookup.DBUM.Sybase.UM.ExclusionList.Trusted. However, you can create new lookups and add them to the configuration lookups to hold resources for which you do not want to perform provisioning and reconciliation operations.

The following is the format of the values stored in these lookups:

Code Key	Decode	Sample Values
Login Name	User ID of a user	Code Key: Login Name Decode: User001
Login Name with the [PATTERN] suffix	A regular expression supported by the representation in the <code>java.util.regex.Pattern</code> class	Code Key: Login Name[PATTERN] To exclude users matching any of the user ID 's User001, User002, User088, then: Decode: User001 User002 User088 To exclude users whose user ID 's start with 00012, then: Decode: 00012* See Also: For information about the supported patterns, visit http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html

[Configuring Resource Exclusion Lists for Sybase](#) describes the procedure to add entries in these lookup definitions.

7.4 Scheduled Jobs for Sybase

When you run the Connector Installer or import the connector XML file, the scheduled jobs are automatically created in Oracle Identity Manager.

This section describes the following topics:

- [Scheduled Jobs for Lookup Field Synchronization for Sybase](#)
- [Attributes for Scheduled Jobs for Sybase](#)
- [Configuring Scheduled Jobs for Sybase](#)

7.4.1 Scheduled Jobs for Lookup Field Synchronization for Sybase

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

The following scheduled jobs are used for lookup field synchronization:

- DBUM Sybase Databases Lookup Reconciliation
- DBUM Sybase DB Groups Lookup Reconciliation
- DBUM Sybase Languages Lookup Reconciliation
- DBUM Sybase Roles Lookup Reconciliation

You must specify values for the attributes of these scheduled jobs. [Table 7-17](#) describes the attributes of these scheduled jobs. The procedure to configure scheduled jobs is described later in the guide.

Table 7-17 Attributes of the Scheduled Jobs for Lookup Field Synchronization

Attribute	Description
Code Key Attribute	<p>Enter the name of the connector or target system attribute that is used to populate the Code Key column of the lookup definition (specified as the value of the Lookup Name attribute).</p> <p>Sample value: <code>__NAME__</code></p> <p>Note: Do not change the value of this attribute.</p>
Decode Attribute	<p>Enter the name of the connector or target system attribute that is used to populate the Decode column of the lookup definition (specified as the value of the Lookup Name attribute).</p> <p>Sample value: <code>__NAME__</code></p>
IT Resource Name	<p>Enter the name of the IT resource for the target system installation from which you want to reconcile user records.</p> <p>Default value: <code>Sybase DB</code></p>
Lookup Name	<p>This attribute holds the name of the lookup definition that maps each lookup definition with the data source from which values must be fetched.</p> <p>Depending on the scheduled job you are using, the default values are as follows:</p> <ul style="list-style-type: none"> • For DBUM Sybase Databases Lookup Reconciliation - <code>Lookup.DBUM.Sybase.Databases</code> • For DBUM Sybase DB Groups Lookup Reconciliation - <code>Lookup.DBUM.Sybase.DBGroups</code> • For DBUM Sybase Languages Lookup Reconciliation - <code>Lookup.DBUM.Sybase.DefaultLang</code> • For DBUM Sybase Roles Lookup Reconciliation - <code>Lookup.DBUM.Sybase.Roles</code>
Object Type	<p>Enter the type of object whose values must be synchronized.</p> <p>Depending on the scheduled job you are using, the default values are as follows:</p> <ul style="list-style-type: none"> • For DBUM Sybase Databases Lookup Reconciliation - <code>__DATABASES__</code> • For DBUM Sybase DB Groups Lookup Reconciliation - <code>__DBGROUPS__</code> • For DBUM Sybase Languages Lookup Reconciliation - <code>__DEFAULTTLANG__</code> • For DBUM Sybase Roles Lookup Reconciliation - <code>__ROLES__</code> <p>Note: Do not change the value of this attribute.</p>
Resource Object Name	<p>Enter the name of the resource object that is used for reconciliation.</p> <p>Default value: <code>Sybase DB User</code></p>

7.4.2 Attributes for Scheduled Jobs for Sybase

The following scheduled jobs are used to reconcile user data in the target resource (account management) mode of the connector:

- DBUM Sybase User Target Reconciliation
- DBUM Sybase User Login Target Reconciliation
- DBUM Sybase Delete User Target Reconciliation
- DBUM Sybase Delete User Login Target Reconciliation

The following scheduled jobs are used to reconcile user data in the trusted source (identity management) mode of the connector:

- DBUM Sybase Trusted Reconciliation
- DBUM Sybase Delete User Trusted Reconciliation

Table 7-18 describes the attributes of the scheduled jobs for user operations.

Table 7-18 Attributes of the Scheduled Jobs for Reconciliation

Attribute	Description
IT Resource Name	Name of the IT resource for the target system installation from which you want to reconcile user records Default value: <code>Sybase DB</code> For DBUM Sybase Trusted Reconciliation, enter the name of the IT resource created for trusted source mode.
Object Type	Type of object you want to reconcile <ul style="list-style-type: none"> • For DBUM Sybase User Target Reconciliation: <code>User</code> • For DBUM Sybase User Login Target Reconciliation and DBUM Sybase Trusted Reconciliation: <code>USERLOGIN</code>
Resource Object Name	Name of the resource object that is used for reconciliation <ul style="list-style-type: none"> • For DBUM Sybase User Target Reconciliation: <code>Sybase DB User</code> • For DBUM Sybase User Login Target Reconciliation: <code>Sybase DB User Login</code> • For DBUM Sybase Trusted Reconciliation: <code>Sybase UserLogin Trusted</code>
Scheduled Task Name	Name of the scheduled job Note: For the scheduled job included with this connector, you must not change the value of this attribute. However, if you create a copy of the task, then you can enter the unique name for that scheduled job as the value of this attribute.

Table 7-19 describes the attributes of the scheduled jobs for delete operations.

Table 7-19 Attributes of the Scheduled Jobs for Delete Operations

Attribute	Description
IT Resource Name	Name of the IT resource for the target system installation from which you want to reconcile user records Default value: <code>Sybase DB</code> For DBUM Sybase Delete Trusted Reconciliation, enter the name of the IT resource created for trusted source mode.
Object Type	Type of object you want to reconcile <ul style="list-style-type: none"> • For DBUM Sybase User Target Reconciliation: <code>User</code> • For DBUM Sybase User Login Target Reconciliation and DBUM Sybase Trusted Reconciliation: <code>USERLOGIN</code>

Table 7-19 (Cont.) Attributes of the Scheduled Jobs for Delete Operations

Attribute	Description
Resource Object Name	Name of the resource object that is used for reconciliation <ul style="list-style-type: none"> • For DBUM Sybase Delete User Target Reconciliation: Sybase DB User • For DBUM Sybase Delete User Login Target Reconciliation: Sybase DB User Login • For DBUM Sybase Delete Trusted Reconciliation: Sybase UserLogin Trusted

7.4.3 Configuring Scheduled Jobs for Sybase

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

See [Scheduled Jobs for Lookup Field Synchronization for Sybase](#) for the scheduled jobs that are part of the connector and for information about their attributes.

To configure a scheduled job:

1. If you are using Oracle Identity Manager release 11.1.1.x:
 - a. Log in to the Administrative and User Console.
 - b. On the Welcome to Oracle Identity Manager Self Service page, click **Advanced** in the upper-right corner of the page.
 - c. On the Welcome to Oracle Identity Manager Advanced Administration page, in the System Management region, click **Search Scheduled Jobs**.
2. If you are using Oracle Identity Manager release 11.1.2.x or later:
 - a. Log in to Oracle Identity System Administration.
 - b. In the left pane, under System Management, click **Scheduler**.
3. Search for and open the scheduled job as follows:
 - a. On the left pane, in the Search field, enter the name of the scheduled job as the search criterion. Alternatively, you can click **Advanced Search** and specify the search criterion.
 - b. In the search results table on the left pane, click the scheduled job in the Job Name column.
4. On the Job Details tab, you can modify the following parameters:

Retries: Enter an integer value in this field. This number represents the number of times the scheduler tries to start the job before assigning the Stopped status to the job.

Schedule Type: Depending on the frequency at which you want the job to run, select the appropriate schedule type.

 **Note:**

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

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

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

 **Note:**

Attribute values are predefined in the connector XML file that you import. Specify values only for those attributes that you want to change.

6. After specifying the attributes, click **Apply** to save the changes.

7.5 Reconciliation from Sybase

Reconciliation involves duplicating in Oracle Identity Manager the creation of and modifications to user accounts on the target system. This section discusses the following topics related to configuring reconciliation:

- [About the Reconciliation Process for Sybase](#)
- [Reconciliation Rules for Sybase](#)
- [Viewing Reconciliation Rules for Sybase](#)
- [Reconciliation Action Rules for Sybase](#)
- [Viewing Reconciliation Action Rules for Sybase](#)
- [Performing Full Reconciliation from Sybase](#)

7.5.1 About the Reconciliation Process for Sybase

This connector can be configured to perform either trusted source reconciliation or target resource reconciliation.

When you configure the target system as a target resource, the connector enables you to create and manage database accounts for OIM Users through provisioning. In addition, data related to newly created and modified target system accounts can be reconciled and linked with existing OIM Users and provisioned resources.

When you configure the target system as a trusted source, the connector fetches into Oracle Identity Manager, data about newly created target system accounts. This data is used to create OIM Users.

 **See Also:**

Reconciliation Based on the Object Being Reconciled in *Oracle Fusion Middleware Administering Oracle Identity Manager* for conceptual information about target resource reconciliation and trusted source reconciliation.

The following is an overview of the steps involved in reconciliation:

1. A SQL query or stored procedure is used to fetch target system records during reconciliation.
2. The scheduled job communicates to connector bundle and runs search operations over it, maps the task attributes to parameters of the reconciliation query or stored procedure, and then runs the query or stored procedure on the target system.
3. Target system records that meet the query or stored procedure criteria are fetched into Oracle Identity Manager.
4. If you have configured your target system as a target resource, then each user record fetched from the target system is compared with existing target system resources assigned to OIM Users. The reconciliation rule is applied during the comparison process.

The next step of the process depends on the outcome of the matching operation:

- If a match is found between the target system record and a resource provisioned to an OIM User, then the database user resource is updated with changes made to the target system record.
- If no match is found, then the target system user record is compared with existing OIM Users. The next step depends on the outcome of the matching operation:

If a match is found, then the target system record is used to provision a resource for the OIM User.

If no match is found, then the status of the reconciliation event is set to No Match Found.

 **Note:**

[Reconciliation Rules for Sybase](#) for information about the reconciliation rule

7.5.2 Reconciliation Rules for Sybase

 **See Also:**

Reconciliation Metadata in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for generic information about reconciliation rules and reconciliation action rules

This section describes the reconciliation rules used by the reconciliation engine for this connector.

The following are the reconciliation rules for target resource reconciliation:

- **Rule name:** DBUM Sybase Target Recon
- **Rule element:** User Login Equals User Name

In these rule elements:

- User Login is the field on the OIM User form.
- User Name is the target system field.

7.5.3 Viewing Reconciliation Rules for Sybase

After you deploy the connector, you can view the reconciliation rule for reconciliation by performing the following steps:



Note:

Perform the following procedure only after the connector is deployed.

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Development Tools**.
3. Double-click **Reconciliation Rules**.
4. Search for the rule name.

7.5.4 Reconciliation Action Rules for Sybase

Reconciliation action rules define that actions the connector must perform based on the reconciliation rules defined for Users.

[Table 7-20](#) lists the action rules for target resource reconciliation.

Table 7-20 Action Rules for Target Resource Reconciliation

Rule Condition	Action
No Matches Found	Assign to Administrator With Least Load
One Entity Match Found	Establish Link
One Process Match Found	Establish Link

7.5.5 Viewing Reconciliation Action Rules for Sybase

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

1. Log in to the Oracle Identity Manager Design Console.
2. Expand **Resource Management**.

3. Double-click **Resource Objects**.
4. Search for and open the resource object.
 - Resource object for Sybase:
Sybase DBUM
5. Click the **Object Reconciliation** tab, and then click the **Reconciliation Action Rules** tab. The Reconciliation Action Rules tab displays the action rules defined for this connector.

7.5.6 Performing Full Reconciliation from Sybase

Full reconciliation involves reconciling all existing user records from the target system into Oracle Identity Manager. After you deploy the connector, you must first perform full reconciliation.

To perform a full reconciliation run, remove (delete) any value currently assigned to the Filter attribute and run one of the following scheduled jobs:

- For Sybase as a target resource: DBUM Sybase User Target Reconciliation and DBUM Sybase User Login Target Reconciliation
- For Sybase as a trusted source: DBUM Sybase Trusted Reconciliation

See [Attributes for Scheduled Jobs for Sybase](#) for more information about these scheduled jobs.

7.6 Provisioning for Sybase

Provisioning involves creating or modifying User and User Login accounts on the target system through Oracle Identity Manager.

This section contains the following topics about provisioning:

- [Guidelines on Performing Provisioning Operations for Sybase](#)
- [Understanding the Provisioning Process for Sybase](#)
- [Configuring Direct Provisioning for Sybase](#)
- [Configuring Request-Based Provisioning for Sybase](#)
- [Switching Between Request-Based Provisioning and Direct Provisioning for Sybase](#)
- [Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x](#)

7.6.1 Guidelines on Performing Provisioning Operations for Sybase

The following are guidelines that you must apply while performing provisioning operations:

- Before you perform provisioning operations, lookup definitions must be synchronized with the lookup fields of the target system. In other words, run the scheduled jobs for lookup field synchronization before provisioning operations.
- Passwords for User and User Login accounts provisioned from Oracle Identity Manager must adhere to the password policy set in the target system.

- The character length of target system fields must be taken into account when specifying values for the corresponding Oracle Identity Manager fields.
- During an update password provisioning operation, ensure that you clear the existing text in the Password field, and then enter the new password.

7.6.2 Understanding the Provisioning Process for Sybase

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

When you install the connector on Oracle Identity Manager, the direct provisioning feature is automatically enabled. This means that the process form is enabled when you install the connector.

The following are types of provisioning operations:

- Direct provisioning
- Request-based provisioning
- Provisioning triggered by policy changes

If you configure the connector for request-based provisioning, then the process form is suppressed and the object form is displayed. In other words, direct provisioning is disabled when you configure the connector for request-based provisioning. If you want to revert to direct provisioning, then see [Switching Between Request-Based Provisioning and Direct Provisioning for Sybase](#).

The following is an overview of the Create User provisioning process in Sybase that is started through direct provisioning:

1. On the Create User page of the Administrative and User Console, the administrator enters the data required for an OIM User account creation.

Suppose the administrator enters the following values for the fields on the Create User page:

- First Name: John
- Last Name: Doe
- User ID: jdoe

An OIM User account is created for John Doe.

2. The administrator selects the resource to be provisioned to the OIM User account that has been created. In this example, the administrator selects the Sybase DB User Login resource.
3. The administrator enters the data required for provisioning the Sybase DB User Login resource. Suppose the administrator wants to create a local user that requires a password to log in to the database. Therefore, the administrator enters the following details on the resource provisioning process form:

- IT Resource: Sybase DB
- Login Name: JDoe
- Password: my_pa55word

- **Full Name:** John Doe
- **Default Database:** example_db
- **Default Language:** example_lang

In addition, the administrator also enters the following values on the process form for granting roles:

- **Role:** 3~JAVA_ADMIN

Then, the administrator provisions Sybase DB User with the following details on the resource provisioning process form:

- **IT Resource:** Sybase DB
- **Login Name:** JDoe
- **Username:** John Doe
- **Password:** pa55word_u
- **Database Group:** example_group
- **Database Name:** example_db

4. From the information available in the IT resource for the target system, the configuration (Lookup.DBUM.Sybase.Configuration) lookup definition is identified. This lookup definition stores configuration information that is used during connector operations.
5. The connector bundle contains the script (Provisioning.queries) required for provisioning operations.
6. The identifiers in the stored procedure are replaced with the input parameters fetched from the query. Then, the stored procedure with actual values is formed.
7. The connector runs the stored procedure on Sybase and creates the jdoe account on the target system. The next step of the process depends on whether the administrator had entered data for granting roles or privileges to the target system account.

If the administrator did not enter any values for granting roles, then the provisioning process ends here. Otherwise, the process continues to the next step.
8. While performing Step 3, the administrator had entered the required data for granting roles to the jdoe account. Therefore, the corresponding query as mentioned in Step 6 is read.
9. The complete stored procedure that must be run to perform the Add role provisioning operation is formed.
10. The input parameters required to run the stored procedure are fetched from the parameter configuration done using the queries in the query files.
11. The identifiers in the stored procedure (formed in Step 9) are replaced with the input parameters fetched from the query. Then, the stored procedure with actual values is formed.
12. The query runs the stored procedure on the target system (Sybase) and grants the role to the jdoe target system account.

7.6.3 Configuring Direct Provisioning for Sybase

In direct provisioning, the Oracle Identity Manager administrator uses the Administrative and User Console to create a target system account for a user.

To provision a resource by using the direct provisioning approach:

1. Log in to the Administrative and User Console.
2. To first create an OIM User before provisioning a database account to the user:
 - a. On the Welcome to Identity Administration page, in the Users region, click **Create User**.
 - b. On the Create User page, enter values for the OIM User fields, and then click the save icon.
3. To search for an existing OIM User to be provisioned:
 - a. On the Welcome to Identity Administration page, search for the user by selecting **Users** from the Search list on the left pane.

Alternatively, in the Users region, click **Advanced Search - User**, provide a search criterion, and then click **Search**.
 - b. From the list of users displayed in the search results, select the OIM User.

The user details page is displayed.
4. From the Action menu, select **Add Resource**. Alternatively, you can click the add resource icon with the plus (+) sign. The Provision Resource to User page is displayed in a new window.
5. On the Step 1: Select a Resource page, select the **Sybase DB User** resource from the list, and then click **Continue**.
6. On the Step 2: Verify Resource Selection page, click **Continue**.
7. On the Step 5: Provide Process Data page, enter the details of the account that you want to create on the target system and then click **Continue**.
8. If you want to provide child data, then on the Step 5: Provide Process Data page for child data, search for and select the child data for the user on the target system and then click **Continue**. Repeat the same step if you have more than one child data and you want to provision them.
9. On the Step 6: Verify Process Data page, verify the data that you have provided and then click **Continue**.
10. The "Provisioning has been initiated" message is displayed. Perform the following steps:
 - a. Close the window displaying the "Provisioning has been initiated" message.
 - b. On the Resources tab, click **Refresh** to view the newly provisioned resource.

If the resource status is Provisioned, then provisioning was successful. If the status is Provisioning, then there may be an error. To verify if there was an error, you can check the resource history.

7.6.4 Configuring Request-Based Provisioning for Sybase

The following sections discuss the steps to be performed to enable request-based provisioning:

- [About Request Based Provisioning for Sybase](#)
- [Enabling Request-Based Provisioning](#)

7.6.4.1 About Request Based Provisioning for Sybase

In request-based provisioning, an end user creates a request for a resource by using the Administrative and User Console. Administrators or other users can also create requests for a particular user. Requests for a particular resource on the resource can be viewed and approved by approvers designated in Oracle Identity Manager.

The following are features of request-based provisioning:

- A user can be provisioned only one resource (account) on the target system.

 **Note:**

Direct provisioning allows the provisioning of multiple database accounts on the target system.

- Direct provisioning cannot be used if you enable request-based provisioning.

7.6.4.2 Enabling Request-Based Provisioning

The following sections provide information about the procedures you must perform to enable request-based provisioning:

 **Note:**

The procedure described in this section is applicable only if you are using Oracle Identity Manager release 11.1.1.x.

- [Approver's Role in Request-Based Provisioning](#)
- [Importing Request Datasets Using Deployment Manager](#)
- [End User's Role in Request-Based Provisioning](#)
- [Enabling the Auto Save Form Feature](#)
- [Running the PurgeCache Utility](#)

7.6.4.2.1 Approver's Role in Request-Based Provisioning

The following are steps performed by the approver in a request-based provisioning operation:

1. Log in to the Administrative and User Console.
2. On the Welcome page, click **Self-Service** in the upper-right corner of the page.
3. On the Welcome to Identity Manager Self Service page, click the **Tasks** tab.
4. On the **Approvals** tab, in the first section, you can specify a search criterion for request task that is assigned to you.

5. From the search results table, select the row containing the request you want to approve, and then click **Approve Task**.

A message confirming that the task was approved is displayed.

7.6.4.2.2 Importing Request Datasets Using Deployment Manager

A request dataset is an XML file that specifies the information to be submitted by the requester during a provisioning operation. These request datasets specify information about the default set of attributes for which the requester must submit information during a request-based provisioning operation.

To import a request dataset XML file by using the Deployment Manager:

1. Log in to the Oracle Identity Manager Administrative and User Console.
2. Click the **Deployment Management** link on the left navigation bar.
3. Click the **Import** link under Deployment Management.
A dialog box for opening files is displayed.
4. Locate and open the request dataset XML file, DBUserManagement-Sybase-Datasets.xml, which is in the xml directory of the installation media.
Details of this XML file are shown on the **File Preview** page.
5. Click **Add File**.
The Substitutions page is displayed.
6. Click **Next**.
The Confirmation page is displayed.
7. Click **Import**.
8. Close the Deployment Manager dialog box.
The request dataset is imported into Oracle Identity Manager.

7.6.4.2.3 End User's Role in Request-Based Provisioning

The following steps are performed by the end user in a request-based provisioning operation:

1. Log in to the Administrative and User Console.
2. On the Welcome page, click **Advanced** in the upper-right corner of the page.
3. On the Welcome to Identity Administration page, click the **Administration** tab, and then click the **Requests** tab.
4. From the Actions menu on the left pane, select **Create Request**.
The Select Request Template page is displayed.
5. From the Request Template list, select **Provision Resource** and click **Next**.
6. On the Select Users page, specify a search criterion in the fields to search for the user that you want to provision the resource, and then click **Search**. A list of users that match the search criterion you specify is displayed in the Available Users list.
7. From the **Available Users** list, select the user to whom you want to provision the account.

If you want to create a provisioning request for more than one user, then from the **Available Users** list, select users to whom you want to provision the account.

8. Click **Move** or **Move All** to include your selection in the Selected Users list, and then click **Next**.
9. On the Select Resources page, click the arrow button next to the Resource Name field to display the list of all available resources.
10. From the Available Resources list, select **Sybase DB User**, move it to the Selected Resources list, and then click **Next**.
11. On the Resource Details page, enter details of the account that must be created on the target system, and then click **Next**.
12. On the Justification page, you can specify values for the following fields, and then click **Finish**.
 - Effective Date
 - Justification

A message confirming that your request has been sent successfully is displayed along with the Request ID.

13. If you click the request ID, then the Request Details page is displayed.
14. To view details of the approval, on the Request Details page, click the **Request History** tab.

7.6.4.2.4 Enabling the Auto Save Form Feature

To enable the Auto Save Form feature:

1. Log in to the Design Console.
2. Expand **Process Management**, and then double-click **Process Definition**.
3. Search for and open the **Sybase DB** process definition.
4. Select the **Auto Save Form** check box.
5. Click the save icon.

7.6.4.2.5 Running the PurgeCache Utility

Run the PurgeCache utility to clear content belonging to the Metadata category from the server cache.

See [Clearing Content Related to the Connector Resource Bundles from the Server Cache](#) for instructions.

The procedure to enable enabling request-based provisioning ends with this step.

7.6.5 Switching Between Request-Based Provisioning and Direct Provisioning for Sybase

If you have configured the connector for request-based provisioning, you can always switch to direct provisioning. Similarly, you can always switch back to request-based provisioning any time. This section discusses the following topics:

- [Switching From Request-Based Provisioning to Direct Provisioning](#)

- [Switching From Direct Provisioning to Request-Based Provisioning](#)

7.6.5.1 Switching From Request-Based Provisioning to Direct Provisioning



Note:

It is assumed that you have performed the procedure described in [Configuring Request-Based Provisioning for Sybase](#).

To switch from request-based provisioning to direct provisioning:

1. Log in to the Design Console.
2. Disable the Auto Save Form feature as follows:
 - a. Expand **Process Management**, and then double-click **Process Definition**.
 - b. Search for and open the **Sybase DB** process definition.
 - c. Deselect the Auto Save Form check box.
 - d. Click the save icon.
3. If the Self Request Allowed feature is enabled, then:
 - a. Expand **Resource Management**, and then double-click **Resource Objects**.
 - b. Search for and open the **Sybase DB User** resource object.
 - c. Deselect the Self Request Allowed check box.
 - d. Click the save icon.

7.6.5.2 Switching From Direct Provisioning to Request-Based Provisioning

To switch from direct provisioning back to request-based provisioning:

1. Log in to the Design Console.
2. Enable the Auto Save Form feature as follows:
 - a. Expand **Process Management**, and then double-click **Process Definition**.
 - b. Search for and open the **Sybase DB** process definition.
 - c. Select the **Auto Save Form** check box.
 - d. Click the save icon.
3. If you want to enable end users to raise requests for themselves, then:
 - a. Expand **Resource Management**, and then double-click **Resource Objects**.
 - b. Search for and open the **Sybase DB User** resource object.
 - c. Select the Self Request Allowed check box.
 - d. Click the save icon.

7.6.6 Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x

To perform provisioning operations in Oracle Identity Manager release 11.1.2.x:

1. Log in to Identity Self Service.
2. If you want to first create an OIM User and then provision a target system account, then:

 **Note:**

See *Creating Users in Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Manager* for more information about creating a user.

- a. In the left pane, under Administration, click **Users**.
The Search Users page is displayed.
 - b. From the Actions menu, select **Create**. Alternatively, you can click **Create** on the toolbar.
 - c. On the Create User page, enter values for the OIM User fields, and then click **Submit**. A message is displayed stating that the user is created successfully.
3. If you want to provision a target system account to an existing OIM User, then:

 **Note:**

See *Searching Users in Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Manager* for more information about searching a user.

- a. In the left pane, under Administration, click **Users**.
The Search Users page is displayed.
 - b. Specify a search criteria to search for the OIM User, and then click **Search**.
 - c. From the list of users displayed in the search results, select the OIM User. The user details page is displayed on the right pane.
4. On the Account tab, click **Request Accounts**.
 5. In the Catalog page, search for and add to cart the application instance (in other words, the account to be provisioned), and then click **Checkout**.
 6. Specify value for fields in the application form and then click **Ready to Submit**.
 7. Click **Submit**.
 8. If you want to provision entitlements, then:
 - a. On the Entitlements tab, click **Request Entitlements**.

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

7.7 Extending the Connector for Sybase

The following sections describe procedures that you can perform to extend the functionality of the connector for addressing your specific business requirements:

Note:

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

- [Modifying Predefined Queries or Creating New Queries for Sybase](#)
- [Configuring Queries to Add Support for Custom Parameters and Lookup Fields for Sybase](#)
- [About Configuring the Connector for Multiple Installations of Sybase](#)
- [About Configuring the Connector for Multiple Trusted Source Reconciliation from Sybase](#)
- [Configuring Validation of Data During Reconciliation and Provisioning for Sybase](#)
- [Configuring Transformation of Data During User Reconciliation for Sybase](#)
- [Configuring Resource Exclusion Lists for Sybase](#)
- [Setting Up Action Scripts for Sybase](#)

7.7.1 Modifying Predefined Queries or Creating New Queries for Sybase

The following sections discuss guidelines that you must apply while modifying the predefined queries or creating new queries:

- [About the Queries for Sybase](#)
- [Syntax of Provisioning Queries for Sybase Database](#)
- [Syntax of Reconciliation Queries for Sybase Database](#)
- [Syntax of List of Values Queries for Sybase Database](#)

7.7.1.1 About the Queries for Sybase

Predefined queries are provided to reconcile target system user records, synchronize lookup field values with Oracle Identity Manager, and for provisioning operations. You can modify the predefined queries or add your own queries.

The query files are included in a JAR file in the bundle directory of the connector installation media. For example, `bundle/org.identityconnectors.dbum-1.0.1116.jar`.

The connector includes the following types of queries:

- **Provisioning Queries**
They are used for create, update, and delete operations. The query file is `scripts/sybase/Provisioning.queries`.
- **List of Values Search Queries**
They are used for reconciliation of lookup definitions. A list of value query operates on a set of values for fields such as profiles, privileges, roles, and tablespaces. The query file is `scripts/sybase/LoVSearch.queries`.
- **Account Search Queries**
They are used for full and delete reconciliation operations. An account search query operates on account and group searches with various conditions. The query file is `scripts/sybase/Search.queries`.

 **Note:**

The stored procedure OUT parameters cannot be configured for write-back on the process form. The returned values cannot be used for any connector operations.

7.7.1.2 Syntax of Provisioning Queries for Sybase Database

The following is the syntax of the queries used for provisioning operations:

```

QUERYID {
  Query="QUERY"
  QueryType="QUERYTYPE"
  Parameters=["PARAM1":"PARAMDEFN1", "PARAM2":"PARAMDEFN2"...]
  ExtensionJoin="EXTENSIONJOIN"
  ExtensionSeparator="EXTENSIONSEPARATOR"
  QueryExtensions=["EXTENSION1", "EXTENSION2"...]
}

```

For example:

```

CREATE_USER {
  Query="CALL sp_adduser {loginName}, {__NAME__}, {databaseGroup}"
  QueryType="STOREDPROC"

  Parameters=["__NAME__": "Type:String", "loginName": "Type:String", "databaseGroup": "Type:String, TAGS:NULLABLE"]
  QueryExtensions=[]
}

```

In this syntax:

- *QUERYID* refers to the unique name of the query.

For example: `CREATE_USER`

- *QUERY* refers to the main query.

For example: `Query="CALL sp_adduser {loginName}, {__NAME__}, {databaseGroup}"`

- *QueryType* refers to the type of the main query, either an SQL query or a stored procedure. The value of *QUERYTYPE* can be `SQL` or `StoredProc`.

For example: `QueryType="STOREDPROC"`

- *Parameters* refers to the list of comma separated parameters and parameter definitions used with the main query, represented by "*PARAM1*":"*PARAMDEFN1*", "*PARAM2*":"*PARAMDEFN2*", and so on.

For example:

```
Parameters=["__NAME__": "Type:String", "loginName": "Type:String", "databaseGroup": "Type:String, TAGS:NULLABLE"]
```

A parameter can have the following attributes:

- *Type* is the type of the parameter.
- *Direction* is the flow of data from the query to or from the parameter. It can have a value of `IN`, `OUT`, or `INOUT`.
- *TAGS* is the enclosure characters that are applied to each parameter before the query is processed. It can have a value of `DOUBLEQUOTES`, `QUOTES`, `UPPERCASE`, or `LOWERCASE`.

If you want to use multiple tags, you must encapsulate the tags in escaped quotes and separate them by commas. However, you must not use `DOUBLEQUOTES` with `QUOTES` or `UPPERCASE` with `LOWERCASE` in the same query.

For example: `"Type:String, TAGS:\ "DOUBLEQUOTES,UPPERCASE\ "`

- *ExtensionJoin* (optional) refers to the operator, represented by *EXTENSIONJOIN*, used to join the main query with query extensions.

For example: `ExtensionJoin=", "`

- *ExtensionSeparator* (optional) refers to the delimiter between query extensions, represented by *EXTENSIONSEPARATOR*.

For example: `ExtensionSeparator=", "`

- *QueryExtensions* (optional) refers to the extensions that must be appended to the main query, represented by *EXTENSION1*, *EXTENSION2*, and so on.

During a provisioning operation, the connector combines all these components to the following query:

```
QUERY PARAM1, PARAM2... [EXTENSIONJOIN [EXTENSION1 EXTENSIONSEPARATOR EXTENSION2 EXTENSIONSEPARATOR...]]
```

For example:

```
CALL sp_adduser {loginName}, {__NAME__}, {databaseGroup}
```

[Table 7-21](#) lists the script selection logic of the provisioning queries:

Table 7-21 Script Section Logic for Sybase Provisioning Queries

Operation	Selection Logic	Query IDs
CREATE	CREATE_ OBJECTTYPE	CREATE_USER CREATE_USERLOGIN
DELETE	DELETE_ OBJECTTYPE	DELETE_USER DELETE_USERLOGIN
ENABLE	ENABLE_ OBJECTTYPE	ENABLE_USERLOGIN
DISABLE	DISABLE_ OBJECTTYPE	DISABLE_USERLOGIN
RESET PASSWORD	OBJECTTYPE_SET_PASSWORD	USERLOGIN_SET_PASSWORD
ADD CHILD VALUES	OBJECTTYPE_UPDATE_ADD_AT TRIBUTE	USERLOGIN_UPDATE_ADD_ROLE LIST
REMOVE CHILD VALUES	OBJECTTYPE_UPDATE_REVOKE _ATTRIBUTE	USERLOGIN_UPDATE_ REVOKE_ROLELIST
UPDATE	OBJECTTYPE_UPDATE_ATTRIBU TE	USERLOGIN_UPDATE_DEFAULTLA NGUAGE USERLOGIN_UPDATE_DEFAULTD ATABASE USERLOGIN_UPDATE_FULLNAME UPDATE_DATABASEGROUP

7.7.1.3 Syntax of Reconciliation Queries for Sybase Database

The following is the syntax of the search queries used during reconciliation operations:

```

QUERYID {
  Query="QUERY"
  QueryType="QUERYTYPE"
  Parameters=["PARAM1":"PARAMDEFN1", "PARAM2":"PARAMDEFN2"...]
  ExtensionJoin="EXTENSIONJOIN"
  ExtensionSeparator="EXTENSIONSEPARATOR"
  QueryExtensions=["EXTENSION1","EXTENSION2"...]
}

```

For example:

```

SEARCH_USER {
  Query="call sp_helpuser()"
  QueryType="StoredProc"
  Parameters=["__UID__": "Type:String,Direction:OUT,ColName:Users_name",
             "databaseGroup": "Type:String,Direction:OUT,ColName:Group_name",
             "loginName": "Type:String,Direction:OUT,ColName:Login_name"]
  QueryExtensions=[]
}

```

In this syntax:

- *QUERYID* refers to the unique name of the query.

For example: `SEARCH_USER`

QUERYID can be one of the following values:

- `SEARCH_USER`
- `SEARCH_USERLOGIN`

- *QUERY* refers to the main query.

For example: `Query="call sp_helpuser() "`

- *QueryType* refers to the type of the main query, either an SQL query, a stored procedure, or a query extension. The value of *QUERYTYPE* can be `SQL`, `StoredProc`, or `QUERYEXTENSION`.

For example: `QueryType="StoredProc"`

- *Parameters* refers to the list of comma separated parameters and parameter definitions used with the main query, represented by "*PARAM1*":"*PARAMDEFN1*", "*PARAM2*":"*PARAMDEFN2*", and so on.

For example:

```
Parameters=["__UID__": "Type:String,Direction:OUT,ColName:Users_name",
"databaseGroup": "Type:String,Direction:OUT,ColName:Group_name",
"loginName": "Type:String,Direction:OUT,ColName:Login_name"]
```

A parameter can have the following attributes:

- *Type* is the type of the parameter.
- *Direction* is the flow of data from the query to or from the parameter. It can have a value of `IN`, `OUT`, or `INOUT`.
- *ColName* is the column name in the target system corresponding to the parameter in the query.
- *ColQuery* is the query used to fetch values for the corresponding query parameter.
- *ExtensionJoin* (optional) refers to the operator, represented by *EXTENSIONJOIN*, used to join the main query with query extensions.

For example: `ExtensionJoin=", "`

- *ExtensionSeparator* (optional) refers to the delimiter between query extensions, represented by *EXTENSIONSEPARATOR*.

For example: `ExtensionSeparator=", "`

- *QueryExtensions* (optional) refers to the extensions that must be appended to the main query, represented by *EXTENSION1*, *EXTENSION2*, and so on.

During a reconciliation operation, the connector combines all these components to the following query:

```
QUERY PARAM1, PARAM2... [EXTENSIONJOIN [EXTENSION1 EXTENSIONSEPARATOR
EXTENSION2 EXTENSIONSEPARATOR...]]
```

For example:

```
call sp_helpuser() {__UID__}, {databaseGroup}, {loginName}
```

7.7.1.4 Syntax of List of Values Queries for Sybase Database

If a search query is performed on account types, such as User Name, then the query is considered as a reconciliation query. If a search query is performed on any other object, then the query is considered as a list of values query.

The following is the syntax of the list of values queries used for lookup field synchronization:

```
OBJECTTYPE = "QUERY"
```

For example:

```
__DEFAULTLANG__="SELECT alias FROM syslanguages"
```

In this syntax:

- *OBJECTTYPE* refers to the lookup field attribute.
For example: `__DEFAULTLANG__`, `__DATABASES__`, `__DBGROUPS__`, and `__ROLES__`.
- *QUERY* refers to the query used for fetching a lookup field attribute.
For example: `SELECT alias FROM syslanguages`

The list of values queries return values that are used as lookup field entries. By default, the connector includes dedicated scheduled job for each lookup definition. To use a custom lookup definition, you must add custom fields in the query file.

7.7.2 Configuring Queries to Add Support for Custom Parameters and Lookup Fields for Sybase

The connector uses preconfigured queries for connector operations such as create, delete, and search. You can add custom parameters and lookup definition fields as per your requirements.

The procedure to add a parameter or a lookup definition field to a query file is discussed in the following sections:

- [Updating the Query Files for Sybase Database](#)
- [Configuring Oracle Identity Manager](#)

7.7.2.1 Updating the Query Files for Sybase Database

To update the query files:

1. If the connector is already installed, run the Oracle Identity Manager Download JARs utility to download the connector bundle JAR file from the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

```
OIM_HOME/server/bin/DownloadJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/DownloadJars.sh
```

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

2. Copy the bundle JAR file in a temporary directory.

Sample JAR file: `bundle/org.identityconnectors.dbum-1.0.1116.jar`

Sample temporary directory: `c:\temp`

3. Run the following command to extract the connector bundle JAR file:

```
jar -xvf org.identityconnectors.dbum-1.0.1116.jar
```

 **Note:**

You can also run the WinZip or WinRAR utility to extract the contents from the JAR file.

4. Delete the bundle JAR file in the temporary directory.
5. Update the value of **ConnectorBundle-Version** in the manifest file, `META-INF/MANIFEST.MF`, to a new value.

For example:

```
ConnectorBundle-Version: 1.0.1117
```

6. Depending on your requirement, update the query files with new parameters as per the query syntax described in [Modifying Predefined Queries](#) or [Creating New Queries for Sybase](#).

For example, if you want to add a new parameter, `CUSTOM_ATTRIBUTE`, to the `CREATE_USER` provisioning query:

- a. Open the provisioning query file in a text editor.

Sample query file:

```
c:\temp\bundle\org.identityconnectors.dbum-1.0.1116\scripts\sybase\Provisioning.queries
```

- b. Add the parameter, `CUSTOM_ATTRIBUTE`, to the `CREATE_USER` query.

The following is a sample updated query:

```
CREATE_USER {
    Query="CALL sp_adduser {loginName} , {__NAME__} , {databaseGroup},
{CUSTOM_ATTRIBUTE}"
    QueryType="STOREDPROC"

    Parameters=["__NAME__": "Type:String", "loginName": "Type:String", "databaseG
roup": "Type:String, TAGS:NULLABLE",
"CUSTOM_ATTRIBUTE": "Type:String, Direction:IN"]
    QueryExtensions=[]
}
```

- c. Save and close the query file.
7. Create a new bundle JAR file that contains the updated manifest file and the provisioning query file as follows:

- a. Open the command prompt and navigate to the temporary directory:

```
c:\temp
```

- b. Run the following command:

```
jar -cvfm org.identityconnectors.dbum-1.0.1117.jar *
```

The new connector bundle JAR name contains the new bundle version.

8. In the case of a remote connector server, copy the new bundle JAR file in the bundles directory of the remote connector server, instead of posting the JAR file to the Oracle Identity Manager database. Skip to Step 10.
9. Run the Oracle Identity Manager Update JARs utility to update the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

If you have installed both the Oracle and Sybase connectors on the same Oracle Identity Manager, then ensure that all third-party JAR files are part of the `/lib` directory in the connector bundle JAR file.

For Microsoft Windows:

```
OIM_HOME/server/bin/UpdateJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UpdateJars.sh
```

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

10. Update the configuration lookup with the new bundle version.

For example, you can update the `Lookup.DBUM.Sybase.Configuration` lookup definition.

7.7.2.2 Configuring Oracle Identity Manager

You can skip this procedure if the parameter you added already exists as a default form field in Oracle Identity Manager.

To configure Oracle Identity Manager for adding a parameter:

1. Log into Oracle Identity Manager Design Console.
2. Create a new version of the process form:
 - a. Expand **Development Tools**.
 - b. Double-click **Form Designer**.
 - c. Search for and open the **UD_DB_SYB_U** process form.
 - d. Click **Create New Version**.

On the Create a new version dialog box, enter a new version in the Label field, and then click the save icon.
3. Add the new field on the process form.
 - a. Click **Add**.

A field is added to the list. Enter the details of the field.

For example, if you are adding the CustomAttribute1 field, enter `UD_DB_SYB_U_CUSTOM1` in the **Name** field and then enter the rest of the details of this field.
 - b. Click the save icon and then click **Make Version Active**.
4. If you are using Oracle Identity Manager release 11.1.2.x or later, then all changes made to the Form Designer of the Design Console must be done in a new UI form as follows:
 - a. Log in to Oracle Identity System Administration.
 - b. Create and active a sandbox.
 - c. Create a new UI form to view the newly added field along with the rest of the fields. See *Creating Forms By Using the Form Designer in Oracle Fusion Middleware Administering Oracle Identity Manager*.
 - d. Associate the newly created UI form with the application instance of your target system. To do so, open the existing application instance for your resource, from the Form field, select the form (created in Step 4.c), and then save the application instance.
 - e. Publish the sandbox as described in *Publishing a Sandbox of Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.
5. Create an entry for the field in the lookup definition for provisioning as follows:
 - a. Expand **Administration**.
 - b. Double-click **Lookup Definition**.
 - c. Search for and open the **Lookup.DBUM.Sybase.UM.ProvAttrMap** lookup definition.
 - d. Click **Add** and enter the Code Key and Decode values for the field.

The Code Key value must be the form field name. The Decode value must be the attribute name on the target system.

For example, enter `Custom Attribute 1` in the **Code Key** field and then enter `CustomAttribute1` in the **Decode** field.

- e. Click the save icon.
6. Create a process task to update the new field Custom Attribute 1 as follows:
 - a. Expand **Process Management**.
 - b. Double-click **Process Definition** and open the **Sybase DB User** process definition.
 - c. Click **Add** and enter the task name, for example, `Custom Attribute 1 Updated`, and the task description.
 - d. In the Task Properties section, select **Conditional** and **Allow Multiple Instances** fields and click the save icon.
 - e. On the Integration tab, click **Add**, and then click **Adapter**.
 - f. Select the **adpSYBASEDBUMUPDATEUSER** adapter, click the save icon, and then click **OK** in the message that is displayed.
 - g. To map the adapter variables listed in this table, select the adapter, click **Map**, and then specify the data given in the following table:

Variable Name	Data Type	Map To	Qualifier	Literal Value
Adapter return value	Object	Response code	NA	NA
attributeName	String	Literal	String	Custom Attribute 1
itRes	String	Literal	String	UD_DB_SYB_U_ITRES
objectType	String	Literal	String	User
processInstanceKey	Long	Process Data	Process Instance	NA

- h. On the Responses tab, click **Add** to add the following response codes:

Code Name	Description	Status
ERROR	Error occurred	R
UNKNOWN	An unknown response was received	R
SUCCESS	Operation completed	C

- i. Click the save icon and then close the dialog box.

7.7.3 About Configuring the Connector for Multiple Installations of Sybase

You might want to configure the connector for multiple installations of the target system. The following example illustrates this requirement:

The London and New York offices of Example Multinational Inc. have their own installations of the target system. The company has recently installed Oracle Identity Manager, and they want to configure Oracle Identity Manager to link all the installations of the target system.

To meet the requirement posed by such a scenario, you can create copies of connector objects, such as the IT resource and resource object.

The decision to create a copy of a connector object might be based on a requirement. For example, an IT resource can hold connection information for one target system installation. Therefore, it is mandatory to create a copy of the IT resource for each target system installation.

With some other connector objects, you do not need to create copies at all. For example, a single attribute-mapping lookup definition can be used for all installations of the target system.

All connector objects are linked. For example, a scheduled job holds the name of the IT resource. Similarly, the IT resource for a target system such as Sybase holds the name of the configuration lookup definition, `Lookup.DBUM.Sybase.Configuration`. If you create a copy of an object, then you must specify the name of the copy in associated connector objects.

 **Note:**

- To reconcile data from a particular target system installation, specify the name of the IT resource for that target system installation as the value of the scheduled job attribute that holds the IT resource name. For example, you enter the name of the IT resource as the value of the IT resource attribute of the scheduled job that you run.
- When you use Identity Self Service to perform provisioning, you can specify the IT resource corresponding to the target system installation to which you want to provision the user.

[Table 7-22](#) lists associations between connector objects whose copies can be created and the other objects that reference these objects. When you create a copy of a connector object, use this information to change the associations of that object with other objects.

 **Note:**

- On a particular Oracle Identity Manager installation, if you create a copy of a connector object, then you must set a unique name for it.
- If you are using Oracle Identity Manager release 11.1.2.x or later, then in addition to the procedure described in this section, you must create an application instance for each IT resource. See [Configuring Oracle Identity Manager Release 11.1.2 or Later](#) for information on creating an application instance.

Table 7-22 Connector Objects and Their Associations

Connector Object	Name	Referenced By	Comments on Creating a Copy
IT resource	Sybase DB	<ul style="list-style-type: none"> • UD_DB_SYB_U (process form) • Scheduled tasks 	Create a copy of the IT resource with a different name.

Table 7-22 (Cont.) Connector Objects and Their Associations

Connector Object	Name	Referenced By	Comments on Creating a Copy
Resource object	Sybase DB User Sybase DB Trusted	All connector operations	<p>It is optional to create a copy of the resource object. If you are reconciling the same set of attributes from all installations of the target system, then you need not create a copy of the resource object.</p> <p>Note: Create copies of the resource object only if there are differences in attributes between the various installations of the target system.</p>
Scheduled Jobs	There are many scheduled jobs for different purposes.	NA	<p>You can use the scheduled jobs with the same names. However, you must update the values of the parameters depending on the target system you want to use.</p>
Process definition	Sybase DB User	NA	<p>It is optional to create a copy of the process definition. If you are reconciling or provisioning the same set of attributes from all installations of the target system, then you need not create a copy of the process definition.</p> <p>Note: Create copies of the process form only if there are differences in attributes between the various installations of the target system.</p>
Process form	UD_DB_SYB_U UD_DB_SYB_L	Sybase DB User and Sybase DB User Login (Process definitions)	<p>It is optional to create a copy of the process form. If you are provisioning the same set of attributes from all installations of the target system, then you need not create a copy of the process definition.</p> <p>Note: Create copies of the process form only if there are differences in attributes between the various installations of the target system.</p>
Child process form	<ul style="list-style-type: none"> • UD_DB_SYB_R 	<ul style="list-style-type: none"> • Sybase DB User (Process definition) • UD_DB_SYB_U (Process form) 	<p>It is optional to create a copy of the child process form. If you are provisioning a new set of child data, then you need to create a copy of the child and parent process forms. Then, assign the newly created child process form to the newly created parent process form.</p>

Table 7-22 (Cont.) Connector Objects and Their Associations

Connector Object Name	Referenced By	Comments on Creating a Copy
Configuration lookup definition for a target system configured as a target resource	Lookup.DBUM.Sybase.Configuration Sybase DB (IT resource)	It is optional to create a copy of the configuration lookup definition. If you are provisioning and reconciling the same set of attributes in all installations of the target system (configured as a target resource), then you need not create a copy of the configuration lookup definition. Note: Create copies of the configuration lookup definition only if there are differences in attributes between the various installations of the target system and you have created a new process form.
Resource object attributes mapping lookup definition (for target resource)	Lookup.DBUM.Sybase.UM.ReconciliationMap NA	It is optional to create a copy of resource object attribute mapping lookup definition. If you are reconciling the same set of attributes in all installations of the target system, then you need not to create a copy of resource object attribute mapping lookup. Note: Create copies of this lookup definition only if there are differences in attributes between the two installations of the target system.

7.7.4 About Configuring the Connector for Multiple Trusted Source Reconciliation from Sybase

Note:

This connector supports multiple trusted source reconciliation.

This section describes an optional procedure. Perform this procedure only if you want to configure the connector for multiple trusted source reconciliation.

The following are examples of scenarios in which there is more than one trusted source for user data in an organization:

- One of the target systems is a trusted source for data about users. The second target system is a trusted source for data about contractors. The third target system is a trusted source for data about interns.
- One target system holds the data of some of the identity fields that constitute an OIM User. Two other systems hold data for the remaining identity fields. In other words, to create an OIM User, data from all three systems would need to be reconciled.

If the operating environment of your organization is similar to that described in either one of these scenarios, then this connector enables you to use the target system as one of the trusted sources of person data in your organization.

The following are the guidelines for configuring multiple trusted source reconciliation:

- By default, trusted source reconciliation is performed for Sybase login entities.
To configure trusted source reconciliation to be performed for user entities, modify the **Resource Object Name** parameter of the reconciliation scheduled jobs to Sybase DB User and **Object Type** from USERLOGIN to User.
- To enable reconciliation for each new trusted source, create a new IT Resource and update the **IT Resource Name** parameter of the scheduled jobs with the name of the new IT Resource.

See *Managing Reconciliation in Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed information about multiple trusted source reconciliation.

7.7.5 Configuring Validation of Data During Reconciliation and Provisioning for Sybase

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

To configure validation of data:

1. Write code that implements the required validation logic in a Java class with a fully qualified domain name (FQDN), such as
`org.identityconnectors.dbum.extension.DBUMValidator`.

This validation class must implement the `validate` method. The following sample validation class checks if the value in the First Name attribute contains the number sign (#):

```
package com.validationexample;

import java.util.HashMap;

public class MyValidator {
    public boolean validate(HashMap hmUserDetails, HashMap
hmEntitlementDetails, String sField) throws ConnectorException {

        /* You must write code to validate attributes. Parent
        * data values can be fetched by using
hmUserDetails.get(field)
        * For child data values, loop through the
        * ArrayList/Vector fetched by
hmEntitlementDetails.get("Child Table")
        * Depending on the outcome of the validation operation,
        * the code must return true or false.
        */

        /*
        * In this sample code, the value "false" is returned if the field
        * contains the number sign (#). Otherwise, the value "true" is
        * returned.
        */
    }
}
```

```

        */
        boolean valid = true;
        String sFirstName = (String) hmUserDetails.get(sField);
        for (int i = 0; i < sFirstName.length(); i++) {
            if (sFirstName.charAt(i) == '#') {
                valid = false;
                break;
            }
        }
        return valid;
    }
}

```

2. Log in to the Design Console.
3. Search for and open one of the lookup definitions (or create a new lookup) listed in [Lookup Definition for Validation of Data in Sybase](#).

For example, **Lookup.DBUM.Sybase.UM.ProvValidations** or **Lookup.DBUM.Sybase.UM.ReconValidation.Trusted**.

For login entities, use the **Lookup.DBUM.Sybase.Login.ProvValidations** lookup definition.

 **Note:**

If you cannot find these lookup definitions, create new lookup definitions.

4. In the **Code Key** column, enter the resource object field name that you want to validate. For example, Username.
5. In the **Decode** column, enter the class name. For example, `org.identityconnectors.dbum.extension.DBUMValidator`.
6. Save the changes to the lookup definition.
7. Search for and open the configuration lookup definition for the target system you use. For example, **Lookup.DBUM.Sybase.UM.Configuration**. For login entities, use the **Lookup.DBUM.Sybase.Login.Configuration** lookup definition.
8. In the **Code Key** column, enter one of the following entries:
 - To configure validation of data for reconciliation:
Recon Validation Lookup
 - To configure validation of data for provisioning:
Provisioning Validation Lookup
9. In the **Decode** column, enter the name of the lookup you updated or created in step 3. For example, **Lookup.DBUM.Sybase.UM.ProvValidations**.
10. Save the changes to the lookup definition.
11. Create a JAR with the class and upload it to the Oracle Identity Manager database as follows:

Run the Oracle Identity Manager Upload JARs utility to post the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

`OIM_HOME/server/bin/UploadJars.bat`

For UNIX:

`OIM_HOME/server/bin/UploadJars.sh`

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

12. Run the PurgeCache utility to clear content related to request datasets from the server cache.
13. Perform reconciliation or provisioning to verify validation for the field, for example, Username.

7.7.6 Configuring Transformation of Data During User Reconciliation for Sybase

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

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

1. Write code that implements the required transformation logic in a Java class with a fully qualified domain name (FQDN), such as `org.identityconnectors.dbum.extension.DBUMTransformation`.

This transformation class must implement the transform method. The following sample transformation class modifies the Username attribute by using values fetched from the `__NAME__` attribute of the target system:

```
package com.transformationexample;

import java.util.HashMap;

public class MyTransformer {
    public Object transform(HashMap hmUserDetails, HashMap
hmEntitlementDetails, String sField) throws ConnectorException {
        /*
         * You must write code to transform the attributes.
         * Parent data attribute values can be fetched by
         * using hmUserDetails.get("Field Name").
        */
    }
}
```

```

        * To fetch child data values, loop through the
        * ArrayList/Vector fetched by hmEntitlementDetails.get("Child
Table")
        * Return the transformed attribute.
        */
        String sUserName = (String) hmUserDetails.get("__NAME__");
        return sUserName + "@example.com";
    }
}

```

2. Log in to the Design Console.
3. Search for and open one of the lookup definitions (or create a new lookup) listed in [Lookup Definitions for Transformation of Data in Sybase](#).

For example, **Lookup.DBUM.Sybase.UM.ReconTransformations**.

For login entities, use the **Lookup.DBUM.Sybase.Login.ReconTransformations** lookup definition.

 **Note:**

If you cannot find these lookup definitions, create new lookup definitions.

4. In the **Code Key** column, enter the resource object field name you want to transform. For example, `Username`.
5. In the **Decode** column, enter the class name. For example, `org.identityconnectors.dbum.extension.DBUMTransformation`.
6. Save the changes to the lookup definition.
7. Search for and open the **Lookup.DBUM.Sybase.UM.Configuration** lookup definition. For login entities, use the **Lookup.DBUM.Sybase.Login.Configuration** lookup definition.
8. In the **Code Key** column, enter `Recon Transformation Lookup`.
9. In the **Decode** column, enter the name of the lookup you updated or created in step 3. For example, **Lookup.DBUM.Sybase.UM.ReconTransformations**.
10. Save the changes to the lookup definition.
11. Create a JAR with the class and upload it to the Oracle Identity Manager database as follows:

Run the Oracle Identity Manager Upload JARs utility to post the JAR file created in Step 7 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

For Microsoft Windows:

`OIM_HOME/server/bin/UploadJars.bat`

For UNIX:

`OIM_HOME/server/bin/UploadJars.sh`

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

12. Run the PurgeCache utility to clear content related to request datasets from the server cache.
13. Perform reconciliation to verify transformation of the field, for example, SimpleDisplayName.

7.7.7 Configuring Resource Exclusion Lists for Sybase

You can specify a list of accounts that must be excluded from reconciliation and provisioning operations. Accounts whose user IDs you specify in the exclusion list are not affected by reconciliation and provisioning operations.

To add entries in the lookup for exclusions during provisioning operations for Oracle Database:



Note:

If the lookup definitions for resource exclusions do not exist, you can create new lookup definitions. See [Lookup Definitions for Exclusion Lists for Sybase](#) for information about the lookup definitions and the format of the entries in these lookups.

To specify user IDs to be excluded during reconciliation operations, add entries in the `Lookup.DBUM.Sybase.UM.ReconExclusions` lookup definition.

1. On the Design Console, expand **Administration** and then double-click **Lookup Definition**.
2. Search for and open the **Lookup.DBUM.Sybase.UM.ProvExclusions** or **Lookup.DBUM.Sybase.UM.ReconExclusions** lookup definition depending on provisioning or reconciliation exclusion lists.

For login entity, use **Lookup.DBUM.Sybase.Login.ProvExclusions** or **Lookup.DBUM.Sybase.Login.ReconExclusions**.

For trusted source reconciliation, use **Lookup.DBUM.Sybase.UM.ExclusionList.Trusted**.

3. Click **Add**.
4. In the Code Key column, enter the resource object field name on which the exclusion list is applied. In the Decode column, enter the corresponding ID of the record to exclude.

For example, if you do not want to provision users with the user ID User001, then you must populate the lookup definition with the following values:

Code Key	Decode
Login Name	User001

 **Note:**

If you want to specify a list of accounts that must be excluded during reconciliation or provisioning, the code key value being specified here must be exactly as the corresponding code key value in the Lookup.DBUM.Sybase.UM.ReconAttrMap lookup definition, or in the Lookup.DBUM.Sybase.UM.ProvAttrMap lookup definition, respectively.

5. If there is more than one user ID to exclude, then in the decode column, enter a list of all user IDs to exclude. Note that each User ID must be separated by a vertical bar (|).

For example, if you do not want to provision users with user IDs User001, User002, and User088 then you must populate the lookup definition with the following values:

Code Key	Decode
Login Name	User001 User002 User088

You can also perform pattern matching to exclude user accounts. You can specify regular expressions supported by the representation in the `java.util.regex.Pattern` class.

 **See Also:**

For information about the supported patterns, visit <http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html>

For example, if you do not want to provision users matching any of the user IDs User001, User002, and User088, then you must populate the lookup definition with the following values:

Code Key	Decode
Login Name[PATTERN]	User001 User002 User088

If you do not want to provision users whose user IDs start with 00012, then you must populate the lookup definition with the following values:

Code Key	Decode
Login Name[PATTERN]	00012*

6. Click the save icon.

7.7.8 Setting Up Action Scripts for Sybase

Learn about action scripts and how to configure them to run before or after the create, update, or delete an account provisioning operations.

This section provides information about the following topics:

- [About Action Scripts for Sybase](#)
- [Configuring Action Scripts for Sybase](#)

7.7.8.1 About Action Scripts for Sybase

Action scripts are scripts that you can configure to run before or after the create, update, or delete an account provisioning operations. For example, you could configure a script to run before every user creation. In another scenario, suppose you have a table called AUDIT_USERLOG where you want to log user creation activities performed only by the connector. Then, you could create and use after create script for adding data to this table after create operation.

Note:

To configure a before or after action, your connector must support running scripts. An exception is Groovy (with target set to **Connector**), which the Identity Connector Framework (ICF) supports by default for all converged connectors.

Every connector should specify which scripting language and which target it supports. This connector supports the following script:

- **shell:** shell script
- **target:** Connector

The target refers to the location where the script is executed. In this case, the script is executed on the same computer (JVM or .NET Runtime) where the connector is deployed. For example, if you deploy the connector on the connector server, the script will be executed on that computer.

That is, if you are using a local framework, the script runs in your JVM. If you are connected to a remote framework, the script runs in the remote JVM or .NET Runtime.

7.7.8.2 Configuring Action Scripts for Sybase

To configure the action:

1. Log in to the Design Console.
2. Search for and open the **Lookup.DBUM.Sybase.UM.Configuration** lookup definition.

For login entities, use the **Lookup.DBUM.Sybase.Login.Configuration** lookup definition.

3. Add the following new values:
 - **Code Key:** Before Create Action Language
 - **Decode:** Enter the scripting language of the script you want to execute
 - Sample values: SQL or STOREDPROC
4. Add these new values:

- **Code Key:** Before Create Action File
- **Decode:** Enter the full path to the file containing the script to be executed (Oracle Identity Manager must be able to access this file.)
- **Example:** /home/scripts/testscript.sql

This script may have a query as follows:

```
INSERT INTO AUDIT_USERLOG VALUES ( {_NAME_}, CURRENT_TIMESTAMP))
```

5. Add these new values:

- **Code Key:** Before Create Action Target
- **Decode:** Connector

6. Save the lookup definition.

Now, this action will be executed every time you create a user. You must configure these three values for each action you want to execute.

8

Configuring the Connector for a JDBC-Based Database

The Database User Management connector is built on a framework designed for JDBC-based connectors. If your target system is a JDBC-based database other than the certified databases listed in [Table 1-1](#), then you can create a connector for your target system by following the instructions given in this chapter.



Note:

In this chapter, MyDatabase has been used as the sample JDBC-based database to explain the procedures.

For Oracle Identity Manager hosted on a Microsoft Windows computer, if you have a previously installed connector, then you must extract the connector bundle zip file again before installing a new connector.

The following sections describe the procedure to create each object of the connector:

- [Target System Attributes and Queries](#)
- [Configuring the Queries](#)
- [Updating the Query Files to the Connector Bundle](#)
- [Configuring the IT Resource](#)
- [Configuring the Process Form](#)
- [Configuring the Resource Object](#)
- [Adding Process Tasks, Assigning Adapters, and Mapping Adapter Variables](#)
- [Adding Attributes for Reconciliation](#)
- [Configuring Lookup Definitions Used During Connector Operations](#)
- [Configuring Scheduled Jobs](#)
- [Configuring Oracle Identity Manager for Request-Based Provisioning](#)
- [Testing the Customized Connector](#)

8.1 Target System Attributes and Queries

This section describes the attributes and the queries of MyDatabase, the sample database used in the procedures in this chapter.

The following table lists the attributes of the database user:

Attribute	Type of Attribute
User Name	String
User Password	String
Database ID	String List of values available in the DBNames table.
Status	String Sample values: ACTIVE, DISABLED
lastModifiedToken	Long

The database users are stored in the MYDBUsers table, which has read-only access. Stored procedures are used to add or modify the users in this table.

The following stored procedures are used in the provisioning queries:

- Call CREATE_USER(usrid, passwd, dbid)
- Call RESET_PASSWD(usrid, passwd)
- Call ENABLE_USER(usrid)
- Call DISABLE_USER(usrid)
- Call DELETE_USER(usrid)
- Call UPDATE_DBID(usrid, dbid)

The following SQL query used to fetch lookup values:

```
Select id from DBNames
```

The following SQL queries are used for reconciliations:

- **Full reconciliation query**

```
Select USRNAME, DBID, Status, lastModifiedToken from MYDBUsers
```
- **Incremental reconciliation query**

```
Select USRNAME, DBID, Status, lastModifiedToken from MYDBUsers where lastModifiedToken > @lastRunToken
```
- **Limited reconciliation query**

This query is similar to a full or incremental reconciliation query, with the filter converted to the WHERE condition. For example, the query for all users with DBID='master' is as follows:

```
Select USRNAME, DBID, Status, lastModifiedToken from MYDBUsers where lastModifiedToken > @lastRunToken AND DBID='master'
```

8.2 Configuring the Queries

This section describes the MyDatabase queries to be created and configured in the connector bundle.

 **See Also:**

The following sections for information about the syntax and samples of the queries used for the certified databases listed in [Table 1-1](#):

- [Modifying the Predefined Queries or Creating New Queries](#)
- [Modifying Predefined Queries or Creating New Queries for MSSQL](#)

1. Create a new file called **Provisioning.queries** and add the following queries in the file:

```
CREATE_USER {
    Query="call CREATE_USER({__NAME__}, {__PASSWORD__}, {dbid})"
    QueryType="STOREDPROC"

    Parameters=["__NAME__": "Type:String", "__PASSWORD__": "Type:GuardedString", "dbid": "Type:String"]
    QueryExtensions=[]
}

DELETE_USER {
    Query="call CREATE_USER({__UID__})"
    QueryType="STOREDPROC"
    Parameters=["__UID__": "Type:String"]
    QueryExtensions=[]
}

ENABLE_USER {
    Query="call ENABLE_USER({__UID__})"
    QueryType="STOREDPROC"
    Parameters=["__UID__": "Type:String"]
    QueryExtensions=[]
}

DISABLE_USER {
    Query="call DISABLE_USER({__UID__})"
    QueryType="STOREDPROC"
    Parameters=["__UID__": "Type:String"]
    QueryExtensions=[]
}

SET_PASSWORD {
    Query="call RESET_PASSWD({__UID__}, {__PASSWORD__})"
    QueryType="STOREDPROC"
    Parameters=["__UID__": "Type:String", "__PASSWORD__": "Type:String"]
    QueryExtensions=[]
}

UPDATE_DBID {
    Query="call UPDATE_DBID({__UID__}, {dbid})"
    QueryType="STOREDPROC"
    Parameters=["__UID__": "Type:String", "dbid": "Type:String"]
    QueryExtensions=[]
}
```

2. Create a new file called **LoVSearch.queries** and add the following query in the file:

```
__DBNAME__=" Select id from DBNames"
```

3. Create a new file called **Search.queries** and add the following queries in the file:

```
SEARCH_USER {
  Query="SELECT {__UID__}, {dbid}, {status}, {lastModified} FROM MYDBUsers
{filter}"
  QueryType="SQL"
  Parameters=["__UID__": "Type:String,Direction:OUT,ColName:USRNAME",
    "dbid": "Type:String,Direction:OUT,ColName:DBID",
    "status": "Type:String,Direction:OUT,ColName:Status",

"lastModified": "Type:long,Direction:OUT,ColName:lastModifiedToken"]
  QueryExtensions=[]
}
```

8.3 Updating the Query Files to the Connector Bundle

This section describes the procedure to update the connector bundle with the MyDatabase query files created in the proceeding section.

To update the query files:

1. If the connector is already installed, run the Oracle Identity Manager Download JARs utility to download the connector bundle JAR file from the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

Note:

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

For Microsoft Windows:

```
OIM_HOME/server/bin/DownloadJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/DownloadJars.sh
```

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

2. Run the following command to extract the connector bundle JAR file:

```
jar -xvf org.identityconnectors.dbum-1.0.1116.jar
```

Note:

You can also run the WinZip or WinRAR utility to extract the contents from the JAR file.

3. Copy the MyDatabase query files to a new directory in the scripts directory of the bundle.

For example: `scripts/mydb`

4. Create a new bundle JAR file that contains the MyDatabase query files as follows:

```
jar uvf org.identityconnectors.dbum-1.0.1116.jar scripts/mydb/*
```

5. If there are any third-party JAR files, copy the JAR files to a new directory, called **lib**, in the bundle. Then, run the following command to update the bundle with the JAR files:

```
jar uvf org.identityconnectors.dbum-1.0.1116.jar lib/*
```

6. If the connector is already installed, run the Oracle Identity Manager Update JARs utility to update the JAR file created in Step 6 to the Oracle Identity Manager database. This utility is copied into the following location when you install Oracle Identity Manager:

 **Note:**

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

If you have installed both the Oracle and MSSQL connectors on the same Oracle Identity Manager, then ensure that all third-party JAR files are part of the `/lib` directory.

For Microsoft Windows:

```
OIM_HOME/server/bin/UpdateJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UpdateJars.sh
```

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

7. If the connector is not installed, then perform the procedure specified in the following sections to deploy the connector:
 - [Preinstallation](#)
 - [Installation](#)

When you deploy the connector, the updated connector bundle will be uploaded to the Oracle Identity Manager database as part of the installation.

8. Create a clone of the connector.

This connector clone will be customized for MyDatabase in the following sections.

 **See Also:**

Cloning Connectors in *Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed information about cloning the connector

9. In the case of a remote connector server, copy the new bundle JAR file in the bundles directory of the remote connector server, instead of posting the JAR file to the Oracle Identity Manager database.

8.4 Configuring the IT Resource

You must specify values for the parameters of the IT resource for MyDatabase as follows:

1. If you are using Oracle Identity Manager release 11.1.1.x:
 - a. Log in to the Administrative and User Console.
 - b. On the Welcome to Oracle Identity Manager Self Service page, click **Advanced** in the upper-right corner of the page.
 - c. On the Welcome to Oracle Identity Manager Advanced Administration page, in the Configuration region, click **Manage IT Resource**.
2. If you are using Oracle Identity Manager release 11.1.2.x or later:
 - a. Log in to Oracle Identity System Administration.
 - b. Create and activate a sandbox. For detailed instructions on creating and activating a sandbox, see *Managing Sandboxes in Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.
 - c. In the left pane, under Configuration, click **IT Resource**.
3. In the IT Resource Name field on the Manage IT Resource page, enter the name of the IT resource, and then click **Search**.
For example: `Oracle DB`
4. Click the edit icon for the IT resource.
5. From the list at the top of the page, select **Details and Parameters**.
6. Specify values for the following parameters of the IT resource. All other parameters of the IT resource will remain unchanged.

[Table 8-1](#) describes the parameters to be updated for MyDatabase.

Table 8-1 IT Resource Parameters for New Database

Parameter	Description
DB Type	This field identifies database type (such as Oracle and MSSQL) and its used for loading respective scripts. Sample value: <code>mydb</code>
JDBC Driver	Specify the value of the JDBC driver class name for MyDatabase.
JDBC URL	Specify the JDBC URL for MyDatabase.
Login Password	Enter the password for the user name of the MyDatabase account to be used for connector operations.
Login User	Enter the user name of the MyDatabase account to be used for connector operations.

7. To save the values, click **Update**.

8.5 Configuring the Process Form

You must update the process form with the attributes of a MyDatabase user. Do not change the IT Resource and Reference ID fields on the process form.

To configure the process form:

1. Log into Oracle Identity Manager Design Console.
2. Create a new lookup definition to hold the Database ID attribute mappings as follows:
 - a. Expand **Administration**.
 - b. Double-click **Lookup Definition**.
 - c. Create a new lookup definition, **Lookup.DBUM.MYDB.DBNames**.

This lookup definition will be empty and will be populated with entries after you run the scheduled jobs for lookup field synchronization.

- d. Click the save icon.
3. Create a new version of the process form:
 - a. Expand **Development Tools**.
 - b. Double-click **Form Designer**.
 - c. Search for and open the **UD_DB_ORA_U** process form.
 - d. Click **Create New Version**.

On the Create a new version dialog box, enter a new version in the Label field, and then click the save icon.

- d. Click the save icon.
4. Add the new fields for the MyDatabase user attributes on the process form.
 - a. Click **Add**.
 - b. Add details for all other attributes as new fields.
 - c. Click the save icon, and then click **Make Version Active**.

A sample screenshot of the process form is as follows:

The screenshot shows the 'Form Designer' window for the table 'UD_DB_ORA_U'. The 'Table Information' section includes the table name, description, and form type (Process). The 'Version Information' section shows the latest version (mydb) and active version (v_11.1.1.5.0). The 'Operations' section includes a dropdown for the current version and buttons for 'Create New Version' and 'Make Version Active'. Below this is a table of columns with the following data:

	Name	Variant Ty...	Len...	Field Label	Field Type	Default Value	Order	App
1	UD_DB_ORA_U_DBID	String	50	Database ID	LookupField		4	
2	UD_DB_ORA_U_ITRES	long		IT Resource	ITResourceLo		1	
3	UD_DB_ORA_U_PASSWORD	String	50	Password	PasswordField		3	
4	UD_DB_ORA_U_RET_ID	String	50	Return Id	DOField		12	
5	UD_DB_ORA_U_USERNAME	String	50	Username	TextField		2	

A sample screenshot for the Database ID attribute is as follows:

The screenshot shows the 'Form Designer' window with the 'Edit Property' dialog open for the 'Database ID' attribute. The dialog displays the following configuration:

- Column Name:** Database ID
- Column Type:** LookupField
- Property Name:** Lookup Code
- Property Value:** Lookup.DBUM.MYDB.DBNames
- Filter Column:** (empty)
- Source:** (empty)
- Field:** (empty)

The background shows the 'Table Information' section and a tree view of the form fields, including 'Database ID (LookupField)' with its property 'Lookup Code = Lookup.DBUM.MYDB.DBNames' highlighted.

8.6 Configuring the Resource Object

You must rename the resource object to **MYDB User** and modify the reconciliation fields as required for MyDatabase. Do not change the IT Resource and Reference ID fields.

 **Note:**

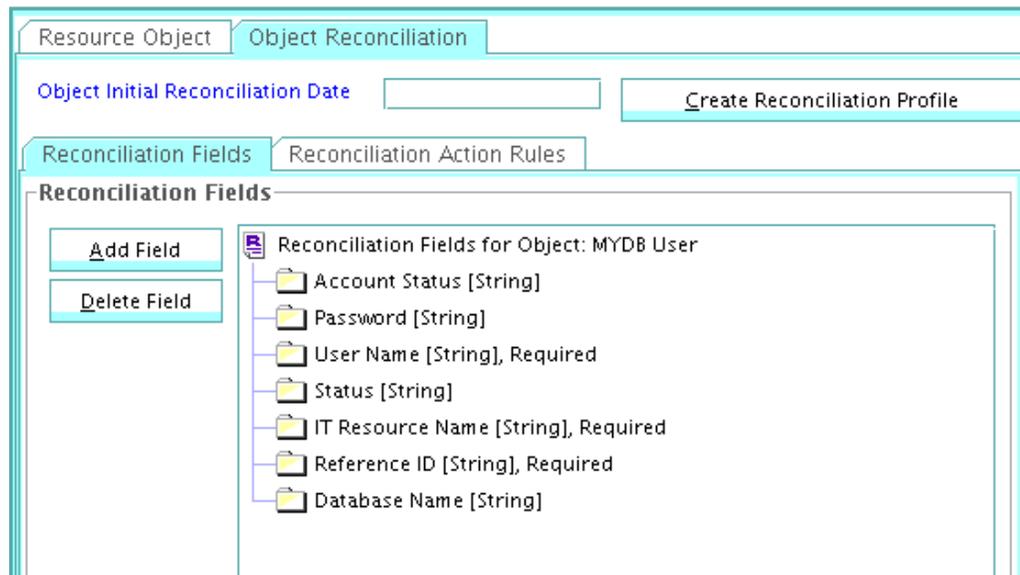
You must remove the process task mappings before removing the reconciliation fields in the resource object.

To rename the resource object:

1. Log in to the Design Console.
2. Expand **Resource Management**, and then double-click **Resource Objects**.
3. Search for and open the resource object of the connector.
For example: Oracle User
4. In the **Name** field, change the name of the resource object to `MYDB User`.
5. If required, you can attach a resource form to the resource object. To do this, double-click the **Table Name** lookup field. From the Lookup dialog box, select the table that represents the form that will be associated with the resource object.
6. To request the resource object for a user, select **Order For User**.
7. Double-click the **Type** lookup field.
From the Lookup dialog box that is displayed, select the classification status **Application** to associate with the resource object.
8. If you want multiple instances of the resource object to be requested for a user or an organization, select the **Allow Multiple** option. Otherwise, go to Step 10.
9. If you want to be able to request the resource object for yourself, select the **Self Request Allowed** option.
10. To provision the resource object for all users, regardless of whether the organization to which the user belongs has the resource object assigned to it, select the **Allow All** check box.
11. Click the save icon.

The resource object is created.

A sample screenshot of the updated resource object with reconciliation fields is as follows:



8.7 Adding Process Tasks, Assigning Adapters, and Mapping Adapter Variables

You must rename the process definition to **MY Database User** and remove the unused process tasks from the process definition.

To integrate the Create User process task with the adpORACREATESETFORM adapter:

1. Expand **Process Management**.
2. Double-click **Process Definition** and open the **MY Database User** process definition.
3. Double-click the **Create User** task to open it.
4. On the Integration tab, click **Add**, and then click **Adapter**.
5. Select the **adpORACREATESETFORM** adapter, click the save icon, and then click **OK** in the message that is displayed.
6. To map the adapter variables listed in this table, select the adapter, click **Map**, and then specify the data given in the following table:

Variable Name	Data Type	Map To	Qualifier	Literal Value
Adapter return value	Object	Literal	NA	Response Code
itRes	String	Literal	String	UD_DB_ORA_U_ITRES
objectType	String	Literal	String	User
processInstanceKey	Long	Process Data	Process Instance	NA
shouldUpdateFlag	String	Literal	NA	False
updateField	String	Literal	NA	NA
updateValue	String	Literal	NA	NA

- On the Responses tab, click **Add** to add the following response codes:

Code Name	Description	Status
ERROR	Error occurred	R
UNKNOWN	An unknown response was received	R
SUCCESS	Operation completed	C

- Click the save icon and then close the dialog box.
- Using the procedure described previously, integrate the **Enable User** process task with the **adpORAENABLEUSERUPDATEFORM** adapter and map the following adapter variables:

Variable Name	Data Type	Map To	Qualifier	Literal Value
Adapter return value	Object	Literal	NA	Response Code
itRes	String	Literal	String	UD_DB_ORA_U_ITRES
objectType	String	Literal	String	User
processInstanceKey	Long	Process Data	Process Instance	NA
shouldUpdateFlag	String	Literal	NA	False
updateField	String	Literal	NA	NA
updateValue	String	Literal	NA	NA

- Using the procedure described previously, integrate the **Disable User** process task with the **adpORADISABLEUPDATEFORM** adapter and map the following adapter variables:

Variable Name	Data Type	Map To	Qualifier	Literal Value
Adapter return value	Object	Literal	NA	Response Code
itRes	String	Literal	String	UD_DB_ORA_U_ITRES
objectType	String	Literal	String	User
processInstanceKey	Long	Process Data	Process Instance	NA
shouldUpdateFlag	String	Literal	NA	False
updateField	String	Literal	NA	NA
updateValue	String	Literal	NA	NA

- Using the procedure described previously, integrate the **Delete User** process task with the **adpORADELETEUSER** adapter and map the following adapter variables:

Variable Name	Data Type	Map To	Qualifier	Literal Value
Adapter return value	Object	Literal	NA	Response Code
itResFieldName	String	Literal	String	UD_DB_ORA_U_ITRES
objectType	String	Literal	String	User
processInstanceKey	Long	Process Data	Process Instance	NA

- Using the procedure described previously, create and integrate the update process task with the **adpORAUPDATEWITHREF** adapter.

The update task names should be named as "*FIELD_NAME* Updated."

For example, the update task for the password field will be **Password Updated**. The following adapter variables must be mapped for the Password Updated task:

Variable Name	Data Type	Map To	Qualifier	Literal Value
Adapter return value	Object	Literal	NA	Response Code
attrName	String	Literal	String	UD_DB_ORA_U_PASSWORD
ITResField	String	Literal	String	UD_DB_ORA_U_ITRES
objectType	String	Literal	String	User
processInstanceKey	Long	Process Data	Process Instance	NA
oldValue	String	Process Data	NA	Password (Field with old value box checked)
newValue	String	Process Data	NA	Password (Field with old value box unchecked)

13. Using the procedure described previously, create and integrate the update process task with the **adpORAUPDATEWITHREF** adapter.

The update task names should be named as "*FIELD_NAME* Updated."

For example, the update task for the Database ID field will be **Database ID Updated**. The following adapter variables must be mapped for the Database ID Updated task:

Variable Name	Data Type	Map To	Qualifier	Literal Value
Adapter return value	Object	Literal	NA	Response Code
attrName	String	Literal	String	UD_DB_ORA_U_DBID
itResFieldName	String	Literal	String	UD_DB_ORA_U_ITRES
objectType	String	Literal	String	User
processInstanceKey	Long	Process Data	Process Instance	NA
oldValue	Long	Process Data	Process Instance	Database ID (Field with old Value box checked)
newValue	Long	Process Data	Process Instance	Database ID (Field with old Value box unchecked)

A sample screenshot of the updated process task is as follows:

Process Definition

Name: MY Database User [Map Descriptive Field](#) [Render Workflow](#)

Type: Provisioning Default Process Auto Pre-populate

Object Name: MYDB User Auto Save Form

Form Assignment

Table Name: UD_DB_ORA_U

Tasks Reconciliation Field Mappings Administrators

	Task ▼	Default Assignee	Event Handler/Adapter	Condi...
Add	1 Create User		adpORACREATESETFORM	<input type="checkbox"/>
Delete	2 Delete User		adpORADELETEUSER	<input checked="" type="checkbox"/>
	3 Disable User		adpORADISABLEUPDATEFORM	<input checked="" type="checkbox"/>
	4 Enable User		adpORAENABLEUSERUPDATEFORM	<input checked="" type="checkbox"/>
	5 Password Updated		adpORAUPDATEWITHREF	<input checked="" type="checkbox"/>
	6 Reconciliation Delete Received			<input checked="" type="checkbox"/>
	7 Reconciliation Insert Received			<input checked="" type="checkbox"/>
	8 Reconciliation Update Received			<input checked="" type="checkbox"/>
	9 Resource Attestation Event Occur			<input checked="" type="checkbox"/>
	10 Service Account Alert			<input checked="" type="checkbox"/>
	11 Service Account Changed			<input checked="" type="checkbox"/>
	12 Service Account Moved			<input checked="" type="checkbox"/>
	13 System Validation			<input type="checkbox"/>
	14 User Attestation Event Occurred			<input checked="" type="checkbox"/>
	15 Username Updated		adpORAPREVENTFUNCTIONALITY	<input checked="" type="checkbox"/>
	16 Database ID Updated		adpORAUPDATEWITHREF	<input checked="" type="checkbox"/>

8.8 Adding Attributes for Reconciliation

After you create the resource object, you must define the attributes on the target resources that must be used for reconciliation. In addition, you must also map these attributes to the corresponding fields on Oracle Identity Manager. Note that the attributes that you add to the resource object are mapped for reconciliation between Oracle Identity Manager and the target system.

A sample screenshot of the attribute mappings for the MY Database User process definition is as follows:

Process Definition

Name: MY Database User

Type: Provisioning Default Process Auto Pre-populate

Object Name: MYDB User Auto Save Form

Form Assignment

Table Name: UD_DB_ORA_U

Reconciliation Field Mappings

Reconciliation Field Mappings for process: MY Database User

- Account Status [String] = UD_DB_ORA_U_LOCK
- User Name [String] = UD_DB_ORA_U_USERNAME, <KEY>, <Case-Insensitive>
- Status [String] = OIM_OBJECT_STATUS
- IT Resource Name [IT Resource] = UD_DB_ORA_U_ITRES, <KEY>
- Reference ID [String] = UD_DB_ORA_U_RET_ID, <KEY>, <Case-Insensitive>
- Database Name [String] = UD_DB_ORA_U_DBID

8.9 Configuring Lookup Definitions Used During Connector Operations

In Oracle Identity Manager, you must configure lookup definitions of the following types that will be used during connector operations:

- Lookup definitions corresponding to lookup fields on the target system
- Lookup definitions that store configuration and other generic information

To modify the values of these lookup definitions:

1. Log in to the Design Console.
2. Expand **Administration**, and then double-click **Lookup Definition**.
3. Update the Lookup.DBUM.Oracle.Configuration lookup definition as follows:
 - a. Search for and open the **Lookup.DBUM.Oracle.Configuration** lookup definition.
 - b. Update the Decode column of **disabledValuesSet** to `Disabled`.
 - c. Update the Decode column of **reservedWordsList** and **unsupportedChars** if you want to add any restrictions on the user inputs.

A sample screenshot of the updated lookup definition is as follows:

Lookup Definition

Code:

Field:

Lookup Type Field Type

Required:

Group:

Lookup Code Information

	Code Key	Decode
1	Bundle Name	org.identityconnectors.dbum
2	Bundle Version	1.0.1116
3	reservedWordsList	"DROP","INSERT","ALTER","CREATE","DELETE","UPDATE","GRANT","TRUNCA"
4	Connector Name	org.identityconnectors.dbum.DBUMConnector
5	User Configuration Lc	Lookup.DBUM.Oracle.UM.Configuration
6	disableValuesSet	"DISABLED"
7	unsupportedChars	"&","-","~","`","\""

4. Update the Lookup.DBUM.Oracle.UM.ProvAttrMap lookup definition as follows:
 - a. Search for and open the **Lookup.DBUM.Oracle.UM.ProvAttrMap** lookup definition.
 - b. Update the provisioning attribute mappings as per MyDatabase. This lookup definition holds user-specific mappings between process form fields (Code Key values) and target system attributes (Decode values) used during provisioning operations (same as the attributes in the Provisioning.queries file).
 - c. If an attribute is of type Lookup, then it has to be tagged with [LOOKUP].

A sample screenshot of the updated lookup definition is as follows:

Lookup Definition

Code:

Field:

Lookup Type Field Type

Required:

Group:

Lookup Code Information

	Code Key	Decode ▼
1	Username	__NAME__
2	Password	__PASSWORD__
3	Return Id	__UID__
4	Database ID[LOOKUP]	dbid

5. Update the Lookup.DBUM.Oracle.UM.ReconAttrMap lookup definition as follows:
 - a. Search for and open the **Lookup.DBUM.Oracle.UM.ReconAttrMap** lookup definition.
 - b. Update the reconciliation attribute mappings as per MyDatabase. This lookup definition holds user-specific mappings between reconciliation attribute names as

specified in the resource object (Code Key values) and target system attributes (Decode values) used during reconciliation operations.

- c. If an attribute is of type Lookup, then it has to be tagged with [LOOKUP].
Do not modify the Reference ID mapping.

A sample screenshot of the updated lookup definition is as follows:

Lookup Definition

Code:

Field:

Lookup Type Field Type

Required:

Group:

Lookup Code Information

	Code Key	Decode
1	Status	__ENABLE__
2	User Name	__UID__
3	Reference ID	__UID__
4	Database Name[LOOKUP]	dbid

6. Click the save icon.



See Also:

[Lookup Definitions Used During Connector Operations for Oracle Database](#) for descriptions of the entries in the lookup definitions

8.10 Configuring Scheduled Jobs

You need scheduled jobs for the following reasons:

- [Configuring Scheduled Jobs for Lookup Field Synchronization](#)

For synchronizing lookup field values with the target system.

- [Configuring Scheduled Jobs for Reconciliation](#)

For fetching data from the target system for reconciliation with Oracle Identity Manager.

8.10.1 Configuring Scheduled Jobs for Lookup Field Synchronization

You need not create scheduled jobs for lookup field synchronization. Instead, you can use the lookup reconciliation scheduled jobs that are shipped with this connector. See [Scheduled Jobs for Lookup Field Synchronization for Oracle Database](#) for more information about these scheduled jobs.

For example, to perform Database ID lookup reconciliation, update the **DBUM Oracle Roles Lookup Reconciliation** scheduled job parameters as follows:

Attribute	Description
Code Key Attribute	Enter the name of the connector or target system attribute that is used to populate the Code Key column of the lookup definition (specified as the value of the Lookup Name attribute). For example: <code>__NAME__</code> Note: Do not change the value of this attribute.
Decode Attribute	Enter the name of the connector or target system attribute that is used to populate the Decode column of the lookup definition (specified as the value of the Lookup Name attribute). For example: <code>__NAME__</code>
IT Resource Name	Enter the name of the IT resource for the target system installation from which you want to reconcile user records. For example: <code>MYDB</code>
Lookup Name	This attribute holds the name of the lookup definition that maps each lookup definition with the data source from which values must be fetched. For example: <code>Lookup.DBUM.MYDB.DBNames</code>
Object Type	Enter the type of object whose values must be synchronized. For example: <code>__DBNAME__</code> (as specified in the <code>LoVSearch.queries</code> file)
Resource Object Name	Enter the name of the resource object that is used for reconciliation. For example: <code>MYDB User</code>

8.10.2 Configuring Scheduled Jobs for Reconciliation

Configure the scheduled jobs for reconciliation as follows:

1. To perform target resource reconciliation, update the **DBUM Oracle User Target Reconciliation** scheduled job as follows:

Attribute	Description
IT Resource Name	Enter the name of the IT resource for the target system installation from which you want to reconcile user records. For example: <code>MYDB</code>
Object Type	Enter the type of object whose values must be synchronized. For example: <code>User</code>
Resource Object Name	Enter the name of the resource object that is used for reconciliation. For example: <code>MYDB User</code>
Incremental Recon Attribute	Time-stamp at which the last reconciliation run started For example: <code>lastModified</code> Note: Do not enter a value for this attribute. The reconciliation engine automatically enters a value for this attribute.

 **Note:**

- For incremental reconciliation, the lastModified time-stamp will be automatically updated after full reconciliation and it will be used for incremental reconciliation runs.
- For limited reconciliation, the filter can be applied in the **Filter** field. You can apply filters on the search attributes, such as `__UID__`, `dbid`, or `status`.

For example, the filter `equalTo('dbid','master')` will add the corresponding WHERE clause in the query. Then, the limited results having `dbid='master'` are processed.

2. To perform target resource delete reconciliation, update the **DBUM Oracle Delete User Target Reconciliation** scheduled job parameters as follows:

Attribute	Description
IT Resource Name	Enter the name of the IT resource for the target system installation from which you want to reconcile user records. For example: MYDB
Object Type	Enter the type of object whose values must be synchronized. For example: User
Resource Object Name	Enter the name of the resource object that is used for reconciliation. For example: MYDB User

8.11 Configuring Oracle Identity Manager for Request-Based Provisioning

 **Note:**

Perform the procedure described in this section only if you are using Oracle Identity Manager release 11.1.1.x and want to configure request-based provisioning.

A request-based provisioning operation involves an end user (a requester) who creates a request for a resource and an approver (an OIM User with the required privileges) who approves the request.

To perform request-based provisioning operations, you must configure a request workflow that suits your requirements. You must update the process form attribute names in request datasets. For complete information on configuring the request workflow.



See Also:

[Configuring Request-Based Provisioning for Oracle Database](#) for a similar procedure for the Oracle Database

8.12 Testing the Customized Connector

As a best practice, you must test the connector after completing all customizations for the new database to ensure that it functions as expected.

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

To run the testing utility, see [Testing the Connector](#).

9

Testing the Connector

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

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

While running the testing utility, the testing utility reads the connectivity information from the IT Resource, lookup definitions from Oracle Identity Manager, and process form data is read from the config.properties file. While running the testing utility, you must ensure that the connector should be deployed and Oracle Identity Manager should be running.

This chapter includes the following procedures:

- [Running the Test Utility for the Oracle Database](#)
- [Running the Test Utility for the MSSQL Database](#)
- [Running the Test Utility for MySQL](#)
- [Running the Test Utility for DB2](#)
- [Running the Test Utility for Sybase](#)

9.1 Running the Test Utility for the Oracle Database

Perform the following steps to run the test utility for Oracle:



Note:

The testing utility might not work for IBM WebSphere Application Server and Oracle WebLogic Server.

1. Set the JAVA_HOME to use JDK1.6, that is the classpath should have the entry to JDK 1.6.
2. If you are using Oracle Identity Manager release 11.1.2.x or later, then include the jrf.jar, jrf-api.jar, and jrf-client.jar files to the classpath.
These JAR files are located in the \$ORACLE_COMMON/modules/oracle.jrf_11.1.1 directory.
3. Copy the third party jar (ojdbc5.jar) in the following directory:
\$DBUM-11.1.1.6.0/test/thirdparty
4. Update OIM_HOME/DBUM-11.1.1.6.0/test/config/oracleconfig.properties using the values specified in the following table. This file is located in the directory.

Name	Description	Sample or Default Value
Other Data		

Name	Description	Sample or Default Value
Action	Enter the type of operation that you want to test: <ul style="list-style-type: none"> CONNECT: To connect to the target system CREATE: To create a user in the target system, that is, provisioning DELETE: To delete the user from the target system ENABLE: To enable the user in the target system DISABLE: To disable the user in the target system ADDROLE: To add/grant role to the user DELETEROLE: To delete/revoke role of the user ADDPRIILEGE: To grant privilege to the user DELETEPRIVILEGE: To delete privilege to the user UPDATEPASSWORD: To update the password 	CONNECT
ObjectType	Oracle supports only one user type	USER
Log Level	Enter one of the following values: <ul style="list-style-type: none"> OFF INFO FINE 	FINE
OIM Login Data		
Security Auth Login Config File	Security Auth Login Config File	<i>OIM_HOME</i> /server/config/authwl.conf
OIM Connection URL	Enter the Oracle Identity Manager connection URL.	t3://OIM_HOST:PORT
Context Factory	Enter the name of the context factory.	weblogic.jndi.WLInitialContextFactory
OIM Admin User	Enter the name of the Oracle Identity Manager user.	xelsysadm
Database IT Resource Name		
IT Resource Name	Enter the name of the IT resource from which connectivity information must be read.	Oracle DB
Process Form Fields and Query Code Keys		
UserID	Enter the ID of the user to be provisioned/created. Note: UserID to be deleted or updated or enable/disable should be already existing on the target.	TestUser
__NAME__	Enter the name of the user to be provisioned. This attribute will be used only while provisioning.	TestUser
__PASSWORD__	Enter the password of the user to be provisioned.	mypassw0r1
authType	Enter one of the following authentication types: <ul style="list-style-type: none"> PASSWORD GLOBAL EXTERNAL Note: This is a mandatory field.	PASSWORD
globalDN	Enter the Global DN of the user. Provide the value for this attribute only when the authentication type is GLOBAL.	Global DN
tablespace	Default Tablespace	tablespace
defaultQuota	Default Tablespace Quota (in MB).	defaultQuota
profile	Profile Name	profile
tempTableSpace	Temporary Tablespace	tempTableSpace

Name	Description	Sample or Default Value
Roles to be added/deleted		
roles~DBRole~__NAME__	Enter the roler name.	CONNECT
roles~DBRole~admin Option	Enter the role with admin option. Note: leave this attribute blank if you do not want to grant admin option.	WITH ADMIN OPTION
Privilege to be added/deleted		
privileges~DBPrivilege~__NAME__	Enter the privilege name.	ALTER ANY ASSEMBLY
privileges~DBPrivilege~adminOption	Enter the privelege with admin option.	WITH ADMIN OPTION

- After you specify values in the oracleconfig.properties file, run the test utility from the following location:

OIM_HOME/DBUM-11.1.1.6.0/test/scripts

For UNIX:

DBUMProvisioningTester.sh

For Microsoft Windows:

DBUMProvisioningTester.bat

- Enter Oracle Identity Manager Administrator's Password on the console when prompted. A message confirming the successful completion of running the utility is displayed.

9.2 Running the Test Utility for the MSSQL Database

Perform the following steps to run the test utility for MSSQL:

- Drop the third party jar (sqljdbc4.jar) in the following directory:
OIM_HOME/DBUM-11.1.1.6.0/test/thirdparty
- If you are using Oracle Identity Manager release 11.1.2.x or later, then include the jrf.jar, jrf-api.jar, and jrf-client.jar files to the classpath.

These JAR files are located in the \$ORACLE_COMMON/modules/oracle.jrf_11.1.1 directory.

- Update *OIM_HOME*/DBUM-11.1.1.6.0/test/scripts/DBUMProvisioningTester.sh and replace oracleconfig.properties with mssqlconfig.properties in the first argument:

```
java -cp "../lib/*:../thirdparty/*"
oracle.iam.connectors.dbum.testutil.prov.ProvisioningUtility
../config/mssqlconfig.properties
../../bundle/org.identityconnectors.dbum-1.0.1116.jar
```

- Update *OIM_HOME*/DBUM-11.1.1.6.0/test/config/mssqlconfig.properties with the required information.

Name	Description	Sample or Default Value
Other Data		
Action	Enter the type of operation that you want to test: <ul style="list-style-type: none"> CONNECT: To connect to the target system CREATE: To create a user in the target system, that is, provisioning DELETE: To delete the user from the target system ENABLE: To enable the user in the target system. This operation is supported for the object type UserLogin only. DISABLE: To disable the user in the target system. This operation is supported for the object type UserLogin only. ADDROLE: To add/grant role to the user. This operation is supported for object type User. DELETEROLE: To delete/revoke role of the user UPDATEPASSWORD: To update the password. This operation is supported only for the object type UserLogin. 	CONNECT
ObjectType	MSSQL supports two user types: USER and USERLOGIN	USERLOGIN
Log Level	Enter one of the following values: <ul style="list-style-type: none"> OFF INFO FINE 	FINE
OIM Login Data		
Security Auth Login Config File	Security Auth Login Config File	<i>OIM_HOME</i> /server/config/authwl.conf
OIM Connection URL	Enter the Oracle Identity Manager connection URL.	t3://OIM_HOST:PORT
Context Factory	Enter the name of the context factory.	weblogic.jndi.WLInitialContextFactory
OIM Admin User	Enter the name of the Oracle Identity Manager user.	xelsysadm
Database IT Resource Name		
IT Resource Name	Enter the name of the IT resource from which connectivity information must be read.	MSSQL DB
Process Form Fields and Query Code Keys		
UserID	Enter the ID of the user to be provisioned/created. Note: UserID to be deleted or updated or enable/disable should be already existing on the target.	TestUser
__NAME__	Enter the name of the user to be provisioned. This attribute will be used only while provisioning.	TestUser
__PASSWORD__	Enter the password of the user to be provisioned.	mypassw0r1
loginName	Enter the login name of the user to be provisioned. This attribute will be used only while provisioning.	TestUser
authType	Enter one of the following authentication types: <ul style="list-style-type: none"> WINDOWS_AUTHENTICATION SQL_SERVER_AUTHENTICATION Note: This is a mandatory field.	SQL_SERVER_AUTHENTICATION

Name	Description	Sample or Default Value
defaultDatabase	Default database, this is an optional attribute.	defaultDatabase
defaultLanguage	Default language, this is an optional attribute.	defaultLanguage
Roles to be added/ deleted		
roles~DBRole~__NA ME__	Enter the role name for the user.	Role

- After you specify values in the `mssqlconfig.properties` file, run the test utility from the following location:

`OIM_HOME/DBUM-11.1.1.6.0/test/scripts`

For UNIX:

`DBUMProvisioningTester.sh`

For Microsoft Windows:

`DBUMProvisioningTester.bat`

- Enter Oracle Identity Manager Administrator's Password on the console when prompted. A message confirming the successful completion of running the utility is displayed.

9.3 Running the Test Utility for MySQL

Perform the following steps to run the test utility for MySQL:

- Drop the third party jar (`mysql-connector-java-5.1.20-bin.jar`) in the following directory:
`OIM_HOME/DBUM-11.1.1.6.0/test/thirdparty`
- If you are using Oracle Identity Manager release 11.1.2.x or later, then include the `jrf.jar`, `jrf-api.jar`, and `jrf-client.jar` files to the classpath.

These JAR files are located in the `$ORACLE_COMMON/modules/oracle.jrf_11.1.1` directory.

- Update `OIM_HOME/DBUM-11.1.1.6.0/test/scripts/DBUMProvisioningTester.sh` and replace `oracleconfig.properties` with `mysqlconfig.properties` in the first argument:

```
java -cp "../lib/*:../thirdparty/*"
oracle.iam.connectors.dbum.testutil.prov.ProvisioningUtility
../config/mysqlconfig.properties
../../bundle/org.identityconnectors.dbum-1.0.1116.jar
```

- Update `OIM_HOME/DBUM-11.1.1.6.0/test/config/mysqlconfig.properties` with the required information.

Name	Description	Sample or Default Value
Other Data		
Action	Enter the type of operation that you want to test: <ul style="list-style-type: none"> CONNECT: To connect to the target system. CREATE: To create a user in the target system, that is, provisioning. DELETE: To delete the user from the target system. UPDATEPASSWORD: To update the password. 	CONNECT

Name	Description	Sample or Default Value
ObjectType	MySQL supports only one user type. Do not modify this entry.	USER
Log Level	Enter one of the following values: <ul style="list-style-type: none"> • OFF • INFO • FINE 	FINE
OIM Login Data		
Security Auth Login Config File	Security Auth Login Config File	<i>OIM_HOME</i> /server/config/authwl.conf
OIM Connection URL	Enter the Oracle Identity Manager connection URL.	t3://OIM_HOST:PORT
Context Factory	Enter the name of the context factory.	weblogic.jndi.WLInitialContextFactory
OIM Admin User	Enter the name of the Oracle Identity Manager user.	xelsysadm
Database IT Resource Name		
IT Resource Name	Enter the name of the IT resource from which connectivity information must be read.	MySQL DB
Process Form Fields and Query Code Keys		
UserID	Enter the ID of the user to be provisioned/created. Note: UserID to be deleted or updated should be already existing on the target.	TestUser
__NAME__	Enter the name of the user to be provisioned. This attribute will be used only while provisioning.	TestUser
__PASSWORD__	Enter the password of the user to be provisioned.	mypassw0r1

5. After you specify values in the `mssqlconfig.properties` file, run the test utility from the following location:

OIM_HOME/DBUM-11.1.1.6.0/test/scripts

For UNIX:

`DBUMProvisioningTester.sh`

For Microsoft Windows:

`DBUMProvisioningTester.bat`

6. Enter Oracle Identity Manager Administrator's Password on the console when prompted. A message confirming the successful completion of running the utility is displayed.

9.4 Running the Test Utility for DB2

Perform the following steps to run the test utility for DB2:

1. Drop the third party jar (`db2jcc.jar`) in the following directory:

OIM_HOME/DBUM-11.1.1.6.0/test/thirdparty
2. If you are using Oracle Identity Manager release 11.1.2.x or later, then include the `jrf.jar`, `jrf-api.jar`, and `jrf-client.jar` files to the classpath.

These JAR files are located in the `$ORACLE_COMMON/modules/oracle.jrf_11.1.1` directory.

- Update `OIM_HOME/DBUM-11.1.1.6.0/test/scripts/DBUMProvisioningTester.sh` and replace `oracleconfig.properties` with `db2config.properties` in the first argument:

```
java -cp "../lib/*:../thirdparty/*"
oracle.iam.connectors.dbum.testutil.prov.ProvisioningUtility
../config/db2config.properties
../../bundle/org.identityconnectors.dbum-1.0.1116.jar
```

- Update `OIM_HOME/DBUM-11.1.1.6.0/test/config/db2config.properties` with the required information.

Name	Description	Sample or Default Value
Other Data		
Action	Enter the type of operation that you want to test: <ul style="list-style-type: none"> CONNECT: To connect to the target system. CREATE: To create a user in the target system, that is, provisioning. DELETE: To delete the user from the target system. ENABLE: To enable the user in the target system. DISABLE: To disable the user in the target system. 	CONNECT
ObjectType	DB2 supports two user types: USER and GROUP	USER
Log Level	Enter one of the following values: <ul style="list-style-type: none"> OFF INFO FINE 	FINE
OIM Login Data		
Security Auth Login Config File	Security Auth Login Config File	<code>OIM_HOME/server/config/authwl.conf</code>
OIM Connection URL	Enter the Oracle Identity Manager connection URL.	<code>t3://OIM_HOST:PORT</code>
Context Factory	Enter the name of the context factory.	<code>weblogic.jndi.WLInitialContextFactory</code>
OIM Admin User	Enter the name of the Oracle Identity Manager user.	<code>xelsysadm</code>
Database IT Resource Name		
IT Resource Name	Enter the name of the IT resource from which connectivity information must be read.	DB2
Process Form Fields and Query Code Keys		
UserID	Enter the ID of the user to be provisioned/created. Note: UserID to be deleted or updated or enable/disable should be already existing on the target.	Dummy_Account
__NAME__	Enter the name of the user to be provisioned. This attribute will be used only while provisioning.	Dummy_Account
userType	Enter the type of the user to be provisioned.	USER

- After you specify values in the `mssqlconfig.properties` file, run the test utility from the following location:

```
OIM_HOME/DBUM-11.1.1.6.0/test/scripts
```

For UNIX:

DBUMProvisioningTester.sh

For Microsoft Windows:

DBUMProvisioningTester.bat

6. Enter Oracle Identity Manager Administrator's Password on the console when prompted. A message confirming the successful completion of running the utility is displayed.

9.5 Running the Test Utility for Sybase

Perform the following steps to run the test utility for Sybase:

1. Drop the third party jar (jconn4.jar) in the following directory:
OIM_HOME/DBUM-11.1.1.6.0/test/thirdparty
2. If you are using Oracle Identity Manager release 11.1.2.x or later, then include the jrf.jar, jrf-api.jar, and jrf-client.jar files to the classpath.
These JAR files are located in the *\$ORACLE_COMMON/modules/oracle.jrf_11.1.1* directory.
3. Update *OIM_HOME/DBUM-11.1.1.6.0/test/scripts/DBUMProvisioningTester.sh* and replace *oracleconfig.properties* with *sybaseconfig.properties* in the first argument:

```
java -cp "../lib/*:../thirdparty/*"
oracle.iam.connectors.dbum.testutil.prov.ProvisioningUtility
../config/sybaseconfig.properties
../../bundle/org.identityconnectors.dbum-1.0.1116.jar
```

4. Update *OIM_HOME/DBUM-11.1.1.6.0/test/config/sybaseconfig.properties* with the required information.

Name	Description	Sample or Default Value
Other Data		
Action	Enter the type of operation that you want to test: <ul style="list-style-type: none"> • CONNECT: To connect to the target system. • CREATE: To create a user in the target system, that is, provisioning. • DELETE: To delete the user from the target system. • ENABLE: To enable the user in the target system. This operation is supported for the object type UserLogin only. • DISABLE: To disable the user in the target system. This operation is supported for the object type UserLogin only. • ADDROLE: To add/grant role to the user. This operation is supported for object type User. • DELETEROLE: To delete/revoke role of the user. This operation is supported for object type User. • UPDATEPASSWORD: To update the password. This operation is supported only for the object type UserLogin. 	CONNECT
ObjectType	Sybase supports two user types: USER and USERLOGIN	USERLOGIN

Name	Description	Sample or Default Value
Log Level	Enter one of the following values: <ul style="list-style-type: none"> OFF INFO FINE 	FINE
OIM Login Data		
Security Auth Login Config File	Security Auth Login Config File	<i>OIM_HOME</i> /server/config/authwl.conf
OIM Connection URL	Enter the Oracle Identity Manager connection URL.	t3://OIM_HOST:PORT
Context Factory	Enter the name of the context factory.	weblogic.jndi.WLInitialContextFactory
OIM Admin User	Enter the name of the Oracle Identity Manager user.	xelsysadm
Database IT Resource Name		
IT Resource Name	Enter the name of the IT resource from which connectivity information must be read.	Sybase DB
Process Form Fields and Query Code Keys		
UserID	Enter the ID of the user to be provisioned/created. Note: UserID to be deleted or updated or enable/disable should be already existing on the target.	TestUser
__NAME__	Enter the name of the user to be provisioned. This attribute will be used only while provisioning.	TestUser
__PASSWORD__	Enter the password of the user to be provisioned.	mypassw0r1
loginName	Enter the login name of the user to be provisioned. This attribute will be used only while user creation.	TestUser TestLogin
databaseGroup	Enter the database group. This attribute will be used only while user creation.	databaseGroup
defaultDatabase	Default database, this is an optional attribute. This attribute will be used only for UserLogin.	defaultDatabase
defaultLanguage	Default language, this is an optional attribute. This attribute will be used only for UserLogin.	defaultLanguage
fullName	Enter full name of the user. This attribute will be used only for UserLogin.	fullName
Roles to be added/deleted		
roles~DBRole~__NAME__	Enter the role name for the user.	Role

- After you specify values in the `mssqlconfig.properties` file, run the test utility from the following location:

OIM_HOME/DBUM-11.1.1.6.0/test/scripts

For UNIX:

`DBUMProvisioningTester.sh`

For Microsoft Windows:

`DBUMProvisioningTester.bat`

6. Enter Oracle Identity Manager Administrator's Password on the console when prompted. A message confirming the successful completion of running the utility is displayed.

10

Known Issues and Workarounds

The following sections describe known issues and workarounds associated with this release of the connector:

- [Connector Issues](#)
- [Oracle Identity Manager Issues](#)
- [Target System Issues](#)

10.1 Connector Issues

The following is an issue and workaround associated with the connector:

10.1.1 Wild Card Expressions not Supported in Privileges for MySQL

For the MySQL database, the connector does not support wild card expressions in privileges for the schema.

There is no workaround available for this issue.

10.2 Oracle Identity Manager Issues

The following are issues and workarounds associated with Oracle Identity Manager:

- [Update of a Child Table does not Work as Expected](#)
- [User Remains in Provisioned Status After a Trusted Delete Reconciliation Run](#)
- [Target Reconciliation Does Not Revoke Roles and Privileges](#)
- [The With Grant Option in Child Data Fails to Appear in Entitlement List](#)
- [The Second Installation of the Connector Fails](#)

10.2.1 Update of a Child Table does not Work as Expected

The following issue is observed while using the connector with Oracle Identity Manager 11g Release 2 BP04 (11.1.2.0.4):

Updating a child table does not work as expected. No provisioning task is triggered.

The fix for this bug is available in Oracle Identity Manager 11g Release 2 BP05 (11.1.2.0.5). This fix is also available as a one-off patch to be applied on Oracle Identity Manager 11g Release 2 BP04 (11.1.2.0.4), from ARU for bug 16053618.

10.2.2 User Remains in Provisioned Status After a Trusted Delete Reconciliation Run

After performing a trusted delete reconciliation operation, a user remains in provisioned status if the user is deleted from Oracle Identity Manager.

As a work around, run the target delete reconciliation scheduled job after running the trusted delete reconciliation task.

For any use case where identity is reconciled authoritatively from a target system to Oracle Identity Manager, and also account daily changes are provisioned from Oracle Identity Manager back to the target system, this behavior will be observed.

The solution is to leverage on the existing infrastructure of account reconciliation and schedule it to run at the optimal frequency so that it follows trusted delete reconciliation scheduled job run. After the account reconciliation is completed, the scheduled job will convert the Account status to Revoked state. Eventually, the identity data maintained inside Oracle Identity Manager would look correct from operational as well as audit perspective.

10.2.3 Target Reconciliation Does Not Revoke Roles and Privileges

If you revoke all the roles and privileges associated with a user from a target system and run target reconciliation, no event or log is generated. The roles and privileges are not revoked from Oracle Identity Manager.

There is no workaround available for this issue.

10.2.4 The With Grant Option in Child Data Fails to Appear in Entitlement List

In Oracle and MySQL databases, With Grant Option in child data does not appear in the entitlements list in Oracle Identity Manager 11g Release 2 (11.1.2.0.0).

There is no workaround available for this issue. However, you can successfully perform a provisioning operation, even if the With Grant Option in child data does not appear in the entitlements list.

10.2.5 The Second Installation of the Connector Fails

For Oracle Identity Manager hosted on a Microsoft Windows computer, installation fails if you try to install the connector again, for a different target system.

As a workaround, you must extract the connector bundle zip file again before installing the connector.

This issue has been fixed in Oracle Identity Manager release 11g R1 PS1 BP09 (11.1.1.5.9).

10.3 Target System Issues

The following are issues and workarounds associated with the target system:

- [Creation of a JDBC Connection Results in an Indefinite Wait Time](#)
- [Incremental Reconciliation Cannot Process Role Updates in MSSQL](#)
- [Relevant Error Log not Provided While Provisioning a Duplicate User](#)
- [Provisioning of Privileges for the Schema not Supported for MySQL](#)

10.3.1 Creation of a JDBC Connection Results in an Indefinite Wait Time

Creating a JDBC connection through Microsoft JDBC Driver 2.0, 3.0, 4.0 CTP 3, or jTDS 1.2.5 driver to a Microsoft SQL Server 2008 R2 database using JDK 1.6.0_29 results in an indefinite wait for a connection.

To work around this issue, use a version later than JDK 1.6.0_29.

10.3.2 Incremental Reconciliation Cannot Process Role Updates in MSSQL

MSSQL incremental reconciliation does not involve role updates. The stored procedure for receiving roles information does not have any timestamp or filter-based query support. Therefore, it is not possible to process role updates using incremental reconciliation in MSSQL.

There is no workaround available for this issue.

10.3.3 Relevant Error Log not Provided While Provisioning a Duplicate User

For the MySQL database, the connector does not provide a relevant error log when you try to provision a duplicate user. This issue is because of the behavior of the MySQL database.

There is no workaround available for this issue.

10.3.4 Provisioning of Privileges for the Schema not Supported for MySQL

For the MySQL database, the connector does not support provisioning of privileges for the schema (information_schema and performance_schema). This issue occurs because a user (for example, root@localhost) who is given ALL on *.* may not have the ALL privilege on the information_schema.* schema.

This is the behavior of the MySQL database.

There is no workaround available for this issue.

A

Configuring Requests



Note:

The procedure described in this section is applicable only if you are using Oracle Identity Manager release 11.1.1.x.

This section describes how to configure a custom request workflow to cater to the specific requirements in your organization. Configuring a custom request workflow involves the following steps:

- [Step 1: Creating a Request Dataset for the Resources](#)
- [Step 2: Uploading Request Datasets into MDS](#)
- [Step 3: Creating SOA Composites Required for Approval](#)
- [Step 4: Registering the SOA Composites in Oracle Identity Manager](#)
- [Step 5: Defining Request Approvals](#)
- [Step 6: Creating Request Templates](#)

The following section describes how to extend the request management operations by using plug-in points:

- [Extending Request Management Operations](#)

A.1 Step 1: Creating a Request Dataset for the Resources

This section describes the request datasets in the following sections:

- [About Request Datasets](#)
- [Default Request Datasets](#)
- [Elements and Properties](#)
- [Sample Request Dataset](#)
- [Child Data](#)
- [Common Request Dataset](#)
- [Configuring Localized Values for Request Datasets](#)

A.1.1 About Request Datasets

Request dataset is an XML definition file that dictates what data needs to be collected during various phases of the request lifecycle. In the request dataset, you can define what attributes need to be submitted by the requester and approver, whether or not an attribute is mandatory, and how UI should render the attribute to the user. Every attribute defined as a part of the dataset is associated with a set of properties that define the behavior of the attributes.

Request dataset also allows you to define additional attributes, which exist only in the context of the request.

Every request is raised by using a request template. Each request template is associated with a request type. Each request type is associated with a request model. There is a one-to-one correspondence between request model and request type. A request model is a specification or configuration that instructs the request engine to work in a specific way for a particular request type. Request models are broadly associated with three types of entities: user, resource, and role. All request models are shipped with Oracle Identity Manager and cannot be configured.

The request model associates the appropriate request dataset when a request is raised for a specific request type. For example, generic Provision Resource request model deals with request for provisioning of any resource objects. Every time a resource is defined, if that resource is expected to be provisioned through a request, then a new dataset can be created with the attributes that need to be collected during request lifecycle. The request datasets that are associated with a nongeneric entity, which is user, has predefined or default request datasets.

Request models that are associated with generic entities does not have default request datasets. For example, the Provision Resource request model is associated with a generic entity, which is resource. For the request models that are associated with non-generic entities, such as user, a default request dataset is available because the user entity has fixed set of default attributes.

A.1.2 Default Request Datasets

[Table A-1](#) lists the request models and the associated default request dataset file names that are shipped with Oracle Identity Manager.

Table A-1 Default Request Datasets Shipped with Oracle Identity Manager

Request Model	Default Dataset File Name	Entity
Create User	CreateUserDataSet.xml	User
Delete User	DeleteUserDataset.xml	User
Enable User	EnableUserDataset.xml	User
Disable User	DisableUserDataset.xml	User
Modify User Profile	ModifyUserDataset.xml	User
Self-Register User	SelfCreateUserDataset.xml	User
Modify Self Profile	ModifyUserDataset.xml	User
Create Role	CreateRoleDataSet.xml	Role
Modify Role	ModifyRoleDataSet.xml	Role
Delete Role	DeleteRoleDataSet.xml.	Role
Assign Roles	AssignRolesDataset.xml	Role
Self Assign Roles	AssignRolesDataset.xml	Role
Self Remove Roles	RemoveRolesDataset.xml	Role
Remove from Roles	RemoveRolesDataset.xml	Role
Provision Resource	No request dataset	Resource
Self-Request Resource	No request dataset	Resource

Table A-1 (Cont.) Default Request Datasets Shipped with Oracle Identity Manager

Request Model	Default Dataset File Name	Entity
Enable Provisioned Resource	No request dataset	Resource
Modify Provisioned Resource	No request dataset	Resource
Self Modify Provisioned Resource	No request dataset	Resource
Disable Provisioned Resource	No request dataset	Resource
De-provision Resource	No request dataset	Resource
Self De-Provision Resource	No request dataset	Resource

Note:

It is not mandatory to have a request dataset for each request model. For example, a request dataset is not required for the De-provision Resource request model because there is no specific data to be collected as a part of request submission or approval. But for the Provision Resource request model, if it involves collection of resource-specific data as a part of request submission or approval, then there must be a dataset defined for that model, which dictates what and how the data is to be collected.

Default request datasets can be customized or configured, such as adding new attributes. The following metadata contains the request model and dataset definitions for default request types:

```
/metadata/iam-features-requestactions/model-data/AssignRolesDataset.xml/metadata/iam-features-requestactions/model-data/CreateRoleDataSet.xml/metadata/iam-features-requestactions/model-data/CreateUserDataSet.xml/metadata/iam-features-requestactions/model-data/DeleteRoleDataSet.xml/metadata/iam-features-requestactions/model-data/DeleteUserDataSet.xml/metadata/iam-features-requestactions/model-data/DeleteUserRequest.xml/metadata/iam-features-requestactions/model-data/DisableUserDataSet.xml/metadata/iam-features-requestactions/model-data/DisableUserRequest.xml/metadata/iam-features-requestactions/model-data/EnableUserDataSet.xml/metadata/iam-features-requestactions/model-data/EnableUserRequest.xml/metadata/iam-features-requestactions/model-data/ModifyRoleDataSet.xml/metadata/iam-features-requestactions/model-data/ModifyUserDataSet.xml/metadata/iam-features-requestactions/model-data/RemoveRolesDataset.xml/metadata/iam-features-requestactions/model-data/ResourceCommonDataset.xml/metadata/iam-features-requestactions/model-data/SelfCreateUserDataSet.xml
```

A.1.3 Elements and Properties

Request dataset is defined by using the following elements and their associated attributes:

- [The request-data-set Element](#)
- [The DataSetValidator Element](#)
- [The AttributeReference Element](#)

- [The Attribute Element](#)

A.1.3.1 The request-data-set Element

The request-data-set element is the root element of the request dataset with the following mandatory attributes:

- **name:** The name of the dataset, such as CreateUserDataSet
- **entity:** The underlying entity, such as user, with which the dataset is associated
- **operation:** The operation associated with the dataset, such as CREATE

The following example shows the request-data-set element:

```
<request-data-set xmlns="http://www.oracle.com/schema/oim/request"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
www.oracle.com/schema/oim/request" name="CreateUserDataSet" entity="User"
operation="CREATE">
```

This root element is shown without any child elements.

To create request datasets for the resource entity, refer to [Table A-2](#) that lists the request dataset name format and operation for each request type.

Table A-2 Request Datasets for Resource Entity

Request Type	Request Dataset Name Format	Operation
Provision Resource	ProvisionResource\${ENTITY-NAME}	PROVISION
Self-Request Resource	ProvisionResource\${ENTITY-NAME}	PROVISION
Modify Provisioned Resource	ModifyResource\${ENTITY-NAME}	MODIFYRESOURC E
Self Modify Provisioned Resource	ModifyResource\${ENTITY-NAME}	MODIFYRESOURC E

The following is a sample dataset tag for the Provision Resource dataset for the E-Business RO resource:

```
<request-data-set xmlns="http://www.oracle.com/schema/oim/request"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
www.oracle.com/schema/oim/request" name="ProvisionResourceE-Business RO"
entity="E-Business RO" operation="PROVISION">
```

Here:

- For name, the ProvisionResource\${ENTITY-NAME} format specified in [Table A-2](#) is replaced by the resource name E-Business RO. Replace \${ENTITY_NAME} with the resource name.
- Specify operation as shown in [Table A-2](#).
- Values for the properties are:
 - name: ProvisionResourceE-Business RO
 - entity: E-Business RO
 - operation: PROVISION

A.1.3.2 The DataSetValidator Element

The `DataSetValidator` element is an optional element in request dataset. It is one of the child elements of the `request-data-set` element that describes the user-defined plug-in details meant for validating the dataset attribute values. The request engine runs the implemented plug-in to validate request data during submission. If the validation is successful, then the request gets created. Otherwise, the request is not created. You must implement the plug-in logic. Each dataset can have only one `DataSetValidator` defined. Following are the attributes:

- **name:** This attribute specifies a logical name of the `DataSetValidator` plug-in.
- **classname:** This attribute specifies the fully qualified name of the implemented plug-in class.

The following example shows the `DataSetValidator` element:

```
<DataSetValidator name="CreateUserDataValidator"  
classname="oracle.iam.requestactions.plugins.datavalidator.CreateUserDataValidator"/>
```

In each default request dataset mentioned in [Table A-1](#), the default `DataSetValidator` element is defined and the corresponding implementation is provided by default. The classname in the `DataSetValidators` in these default request datasets can be changed to a customized class to have customized validation.



See Also:

["Validating Request Data"](#) for information about custom validation of request data after submission.

A.1.3.3 The AttributeReference Element

This child element is used to define the entity attributes at request end that take part in data-flow between request dataset and underlying entity attribute or process-form fields. All the data corresponding to `AttributeReference` are collected as request data at various stages of the request life cycle based on the configuration.

Multiple `AttributeReference` elements can be provided in a dataset, one for each attribute.

This section contains the following topics:

- [Mandatory Properties](#)
- [Optional Properties](#)
- [The PrePopulationAdapter Element](#)
- [The lookupValues Element](#)
- [The lookupQuery Element](#)

A.1.3.3.1 Mandatory Properties

The following mandatory properties are used to configure `AttributeReference`:

- **name:** This is the unique name to identify the element. The request refers to an `AttributeReference` by using this name. Its value is of type `String`.
- **attr-ref:** This is the mapping property between the data set value and process form field or underlying entity attribute. For example, the definition `<AttributeReference name="Organization" attr-ref="act_key">` in the `Create User` dataset specifies that the request collects the data as `Organization` corresponding to the new user being created and gets populated in the `act_key` data field of the user entity.

Similarly, with the `AttributeReference name="Domain" attr-ref="domain"` definition associated with a `Provision Email` resource dataset, the request collects the data as `Domain` corresponding to the resource being provisioned, and the data is populated to the `domain` field of the process form. Therefore, for provision resource models, `attr-ref` attribute value must be the same as the value of the field label (`SDC_LABEL`) of the process form. For other models, the `attr-ref` attribute value must be the entity attribute of the underlying entity defined by the feature. Its value is of type `String`.

`DataFlows` is required to map the request dataset attributes to the underlying entity attributes. For example, in some request models, such as `provision resource`, you might need to define a data flow mapping between the request dataset and the process form data fields. The data flow mapping can be achieved by specifying the following in the dataset:

```
<AttributeReference name="ATTRIBUTE_NAME" attr-ref="DATA_FIELD_NAME_IN_PROCESS_FORM" available-in-bulk="false" />
```

For user-based and role-based request datasets, `attr-ref` value is the attribute name specified in user and role entity definitions. For resource-based request models, `attr-ref` value must be the label name of the process form attribute. But for a child form, `attr-ref` value must be the child table name.

- **type:** This property specifies the data type of the value. For example, `type="String"` for the `First Name` attribute specifies that the `First Name` field in the `Create User` request UI accepts `String` type input. The supported data types are:
 - `Byte`
 - `Double`
 - `Integer`
 - `String`
 - `Short`
 - `Long`
 - `Date`
 - `Boolean`
 - `ByteArray`
 - `Clob`

 **Note:**

Out of all the supported types, only Clob data is not displayed in the request management UI.

Attributes of type Clob must not be made as approver-only.

- **widget:** This property is used to specify how the data attribute is to be displayed in the UI at the time of data collection. The value of this property is of type String. The following widgets are supported:
 - **text:** Specifies a text box that allows the user to enter text in a single line. For example, `widget="text"` for the First Name attribute specifies that the First Name field in the Create User UI is a text box.
 - **date:** Specifies a date and time type field. For example, `widget="date"` for the Start Date attribute specifies that the Start Date field in the Create User UI accepts a date as input.
 - **entity:** Specifies an entity type field. When you specify `widget="ENTITY"`, you must specify a value for the `entity-type` property, such as `entity-type="ORGANIZATION"`. This means that the Organization field in the Create User request UI provides the organization lookup from which you can search and select an organization that is present in Oracle Identity Manager. When `widget=ENTITY`, then the value of `entity-type` must be `USER`, `ORGANIZATION`, or `ROLE`.

 **Note:**

You must specify a value for the `attr-ref` element, which is the attribute name for organization in the user entity definition. For example, `attr-ref="act_key"` in the User.xml, which contains the entity definition of the user entity.

- **textarea:** Specifies a large text field for entering multiple lines of text.
- **dropdown:** Specifies a List of Values (LOV). When you specify `widget="dropdown"`, you must specify values for a list of `lookupValues encoded-value` and `decoded-value` elements, such as `lookupValues encoded-value="End User" decoded-value="Identity Only"` and `lookupValues encoded-value="End-User Administrator" decoded-value="End-User Administrator"`. This means that the User Type field in the Create User request UI is displayed as an LOV from which the user types **Identity Only** or **End-User Administrator** can be selected. However, if the `lookup-code` property is defined, as described later in this chapter, then you do not have to specify `lookupValues`.

The list of values can come from either of the following:

- Static list of `lookupValues` specified in the dataset itself. For example:

```
lookupValues encoded-value="End User" decoded-value="Identity Only" and
lookupValues encoded-value="End-User Administrator" decoded-value="End-User
Administrator"
```

- List of lookup values based on the `lookup-code` property defined, as described later in this chapter.

- **radio:** Specifies a radio button.
- **checkbox:** Specifies a checkbox field. This widget can be associated with attribute references only with Boolean type.
- **lookup:** Specifies a lookup field that allows you to select a value from a large number of values. If this is used, then the lookup-code property must be specified.
- **lookup-query:** Specifies a search and select widget that is associated with the lookupQuery element.
- **itresource-lookup:** Specifies a search and select widget that is associated with an IT resource and shows available IT resource instances. For more information about this widget, see the example listed in [Child Data](#).
- **length:** This attribute specifies the length of the data value. For example, `length="80"` for the First Name attribute specifies that the First Name field in the Create User request UI accepts an input of maximum 80 characters. Its value is of type positive integer.
- **available-in-bulk:** Its value is of type Boolean. This property indicates whether or not the attribute reference is to be displayed during bulk request creation.

You can always hide the fields related to single user context, such as first name, user ID, and password, from the request dataset by marking these fields as `available-in-bulk="false"` in a bulk request scenario. This is because the bulk request is applicable for multiple users and the single user fields does not make sense to be displayed on the request UI. For provisioning requests, these single user fields can be populated by the prepopulation adapters filling the process forms directly. If an attribute is specified as `available-in-bulk="false"`, then that attribute cannot be made mandatory. If you create request datasets by using prepopulation adapters, then you cannot have mandatory constraints assigned to any of these single user fields such as first name, user ID, and password. If you want to assign mandatory constraints to single user fields for usability when a single user is requesting for the resource by using self service, then use the `PrePopulationAdapter` element in the request dataset for populating user specific data. See "[The PrePopulationAdapter Element](#)" for information about using the `PrePopulationAdapter` element in request datasets.

A.1.3.3.2 Optional Properties

The following optional attributes can be used to configure `AttributeReference`:

- **required:** This is a flag property to indicate that data value must be supplied at the time of request submission. Value is of type Boolean. If this property is not specified, then the default value taken is false.

Any attribute reference for which the corresponding `attr-ref` field is mandatory in the underlying entity, must be specified as `required="true"`. For example, `Organization` is a mandatory attribute in the `user` entity. Therefore, the corresponding attribute reference in `SelfCreateUserDataset.xml` or `CreateUserDataSet.xml` is specified as `required="true"` reflecting that this field is mandatory in underlying entity as well.

- **Masked:** This is a flag property used to specify if the data value is masked. If the value is set to masked, the request engine always displays it as asterisks. Value is of type Boolean. If this property is not specified, then the default value taken is false.

- **approver-only:** This is a flag property used to specify that data value has to be supplied, edited by approver. By setting this flag, requester is never allowed to supply the corresponding data value. Value is of type Boolean. If this property is not specified, then the default value taken is "false".

If an attribute specified as required="true" and approver-only="true", then approver has to provide the value for this attribute before approving the request.

The approval process does not support adding child table data, and therefore, only one value can be provided for a child table (multivalued) field during the approval process.

- **entity-type:** This property is used to associate an entity from which supported data value is derived for selection from the request UI. For example:

```
<AttributeReference name="Organization" attr-ref="act_key" available-in-bulk="false" type="Long" length="20" widget="ENTITY" required="true" entity-type="ORGANIZATION"/>
```

With this definition, UI displays a lookup widget by using which user can search and select an organization in Oracle Identity Manager.

If entity-type property is defined, then the widget must be defined as ENTITY, and UI displays a lookup widget by using which user can search and select an entity.

- **lookup-code:** This property is used to associate available LKU/LKV values as supported data based on a defined lookup-code. Example:

```
<AttributeReference name="Responsibility Name" attr-ref="Name" type="String" length="30" widget="lookup" required="false" available-in-bulk="true" lookup-code="Oracle.Responsibility.Name"/>
```

This definition renders all the encoded/decoded values for the lookup-code Oracle.Responsibility.Name.

If the lookup-code property is defined, then widget can be defined as lookup, and UI displays a lookup widget by using which user can search and select a lookup value.

If lookup-code is defined, the widget can be dropdown as well. For example:

```
<AttributeReference name="Role" attr-ref="Role" available-in-bulk="false" type="String" length="20" widget="dropdown" lookup-code="Lookup.Users.Role" required="true"/>
```

Here, the User types will be displayed as a dropdown, and user can select any of the values.

If the lookup code is associated with limited number of values, then you can use dropdown. But if the lookup code is associated with large number of values, then use the lookup widget, which allows search and selection of value.

- **itresource-type:** This property is used to associate available itresource instance for a defined itresource-type. Example:

```
<AttributeReference name="Server" attr-ref="Server Name" type="String" widget="itresource-lookup" required="true" itresource-type="EBIZServer" available-in-bulk="true" length="20"/>
```

This definition renders all IT resource instances for the EBIZServer itresource-type.

If the itresource-type property is defined, then widget must be defined as itresource-lookup, and UI displays a lookup widget by using which user can search and select an itresource instance.

- **primary:** This is a flag property used to specify if the dataset attribute can have more than one value. This flag can be set for a dataset attribute only in the context of child table. See "[Child Data](#)" for more information about the primary property.
- **mls:** This is a flag property used to specify if the dataset attribute is of type Multi-Language Support (MLS). Value is of type Boolean. If this property is not specified, then the default value taken is false.
- **entitlement:** This is a flag property used to specify if the dataset attribute is of type entitlement. Value is of type Boolean. If this property is not specified, then the default value taken is false.
- **hidden:** This is a flag property used to specify if the data value is hidden from the approver. This data value is not visible only to the approver but data can be collected from the requester at the time of submitting the request or by other means. Value is of type Boolean. If this property is not specified, then the default value taken is false.

A.1.3.3.3 The PrePopulationAdapter Element

This child element is used to define an associated Oracle Identity Manager plug-in class that helps in generating data values for the corresponding attribute. Each attribute can have a maximum of one PrePopulationAdapter element associated for an AttributeReference definition. An attribute value is prepopulated during request creation from UI with the value returned by the prepopulation adapter plug-in. The attributes are:

- **name:** This attribute is used to specify a logical name of the adapter.
- **classname:** This attribute is used to specify the fully qualified classname of the plugin class.



Note:

Even if a request dataset attribute is configured with a PrePopulationAdapter, its values can be restricted in a request template. As a result, prepopulation does not happen and the values restricted in the template are displayed in the request creation UI.

The following example shows how to associate PrePopulateAdapter for an AttributeReference:

```
<AttributeReference name="Organization" attr-ref="act_key" available-in-bulk="false" type="Long" length="20" widget="ENTITY" required="true" entity-type="ORGANIZATION"/>
<PrePopulationAdapter name="prepopulateOrg"
classname="my.sample.package.SamplePrePopulateOrg" />
</AttributeReference>
```

The my.sample.package.SamplePrePopulateOrg class must be registered as a plug-in with Oracle Identity Manager.

The following example shows a sample data set for provisioning the Active Directory (AD) resource, in which prepopulation is used:

```
<?xml version="1.0" encoding="UTF-8"?>
<request-data-set xmlns="http://www.oracle.com/schema/oim/request"
```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
www.oracle.com/schema/oim/request" name="ProvisionResourceAD" entity="AD"
operation="PROVISION">
<AttributeReference name="Domain" attr-ref="domain" available-in-bulk="true"
type="String" length="20" widget="text">
<PrePopulationAdapter classname="oracle.iam.request.DomainPrepopulateAdapter"/>
</AttributeReference>
<AttributeReference name="Login" attr-ref="login" available-in-bulk="true"
type="String" length="20" widget="text"/>
<AttributeReference name="Organization" attr-ref="organization" available-in-
bulk="true" type="String" length="20" widget="text" required="true">
<PrePopulationAdapter name="org" classname="oracle.iam.request.OrgPrepopulateAdapter"/>
</AttributeReference>
<AttributeReference name="EmployeeType" attr-ref="EmployeeType" available-in-
bulk="true" approver-only="true" type="String" length="20" widget="text"
required="true">
</AttributeReference>
<AttributeReference name="Role" attr-ref="role" available-in-bulk="true" type="String"
length="20" widget="text"><AttributeReference name="RoleName" attr-ref="role" available-
in-bulk="true" type="String" length="20" widget="text" entitlement="true">
</AttributeReference>
<AttributeReference name="Description" attr-ref="description" available-in-bulk="true"
type="String" length="20" widget="text">
</AttributeReference>
</AttributeReference>
</request-data-set>

```

Here, Role is a child form with child attributes RoleName and Description.

The dataset for provisioning AD resource shows that the Organization attribute has a prepopulation adapter associated with it. The organization attribute value will be prepopulated during request creation with the value returned by the prepopulation adapter plug-in.

The value returned by the prepopulate method of the plug-in must be of type corresponding to the type configured in the request dataset. For example, for the Organization attribute listed in the example documented in [Child Data](#), the prepopulate method of OrgPrepopulateAdapter returns a value of type java.lang.String because the type for Organization attribute is configured as String in the dataset.

A.1.3.3.4 The lookupValues Element

The lookupValues element, which is a child element of AttributeReference, is used to define a set of allowable data values for an entity attribute associated with the AttributeReference definition. The attributes of this element are:

- **decoded-value:** This is the data value that is shown to the requester during request creation from UI.
- **encoded-value:** This is the actual data value stored in the request data, and is used for the dataflow. Based on the selected decoded value from the request creation UI, the corresponding encoding value is stored in the request data.

The following sample code snippet shows the AttributeReference that uses lookupValues for the User Type entity attribute:

```

<AttributeReference name="User Type" attr-ref="Xellerate Type" available-in-
bulk="false" type="String" length="30" widget="dropdown" required="true">
<lookupValues encoded-value="End-User Admin" decoded-value="End-User Administrator"/>
<lookupValues encoded-value="Identity" decoded-value="Identity"/>
<lookupValues encoded-value="End-User" decoded-value="End-User"/>
</AttributeReference>

```

The User Type attribute can have one of the three possible values: End-User Admin, Identity Only, and End-User. But the corresponding decoded values are displayed in the dropdown list to the requester at the time of data collection: End-User Administrator, Identity, and End-User respectively. The encoded value is populated into the mapped entity attribute fields as a part of data flow.

A.1.3.3.5 The lookupQuery Element

This child element of AttributeReference used to derive a set of data value dynamically based on a SQL. Request UI shows all the values based on defined lookupQuery in a lookup widget.

```
<AttributeReference name="adminlogin" attr-ref="adminlogin" type="String"
length="20" widget="lookup-query" available-in-bulk="true">
  <lookupQuery lookup-query="SELECT USR_KEY as UKEY, USR_LOGIN as
ULOGIN FROM TEMP_USR where USR_TYPE='$Form Data.admintype'" display-
field="ULOGIN" save-field="UKEY"/>
</AttributeReference>
```

In this example, user key and user login is queried from a table temp_usr based on a SQL query. Following are the properties of this element:

- **lookup-query:** This property value is a generic SQL query supported by the Oracle Identity Manager database. This query can be dependent on another attribute reference of the same dataset. In the example, there is a reference to '\$Form Data.admintype'. This means that the attribute reference 'adminlogin' depends on the attribute reference 'admintype'. The value provided by requester to attribute 'admintype' is used for fetching values for the attribute 'adminlogin' in lookup.
- **display-field:** This property value is one of the alias name from the selected column that needs to be shown to the end-user in the UI attribute, after user selects a value from lookup-widget.
- **save-field:** This property value is one of the alias name from the selected column that needs to be saved internally to the system, after user selects a value from lookup-widget.

The display-field and save-field can be same for an UI attribute.



Note:

In the lookup query, it is mandatory to have aliases for the columns that are used as save-field and display-field.

A.1.3.4 The Attribute Element

As part of request creation, collection of data that does not refer to the underlying entity might be required. The Attribute element can be used to achieve this. An Attribute defined in the request dataset does not require mapping to the underlying entity. These attributes do not require any mapping, and therefore, can be defined in the following way:

```
<Attribute name="ATTRIBUTE_NAME" length="10" type="integer" widget="text"
available-in-bulk="false"/>
```

The attributes are shown in the request details. These can be viewed by the approver so that these can be used for approval decisions.

The Attribute element is similar to the AttributeReference element with a difference. The Attribute element data values are available only in context of request and cannot take part in dataflow. All other properties available with AttributeReference are also available with attribute, except attr-ref attribute.

A.1.4 Sample Request Dataset

The data that needs to be collected by the request is defined in the request dataset XML file. The following is the sample XML code for the Create User dataset:

```
<?xml version="1.0" encoding="UTF-8"?>
<request-data-set xmlns="http://www.oracle.com/schema/oim/request" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
    xsi:schemaLocation="http://www.oracle.com/schema/oim/request" name="CreateUserDataSet"
entity="User" operation="CREATE">
    <DataSetValidator name="CreateUserDataValidator"
classname="oracle.iam.requestactions.plugins.datavalidator.CreateUserDataValidator"/>
    <AttributeReference name="First Name" attr-ref="First Name" available-in-bulk="false" type="String"
length="80"
        widget="text" required="false" mls="false"/>
    <AttributeReference name="Middle Name" attr-ref="Middle Name" available-in-bulk="false"
type="String" length="80"
        widget="text" required="false" mls="false"/>
    <AttributeReference name="Last Name" attr-ref="Last Name" available-in-bulk="false" type="String"
length="80"
        widget="text" required="true" mls="false"/>
    <AttributeReference name="User Login" attr-ref="User Login" available-in-bulk="false" type="String"
length="256"
        widget="text" required="false"/>
    <AttributeReference name="Password" attr-ref="usr_password" available-in-bulk="false" type="String"
length="128"
        widget="text" required="false" masked="true"/>
    <AttributeReference name="Password Generated" attr-ref="Password Generated" available-in-
bulk="false" type="Clob" length="1"
        widget="text" required="false"/>
    <AttributeReference name="Organization" attr-ref="act_key" available-in-bulk="false" type="Long"
length="256"
        widget="ENTITY" required="true" entity-type="ORGANIZATION"/>
    <AttributeReference name="User Type" attr-ref="Xellerate Type" available-in-bulk="false"
type="Boolean" length="30"
        widget="checkbox" required="false"/>
    <AttributeReference name="Role" attr-ref="Role" available-in-bulk="false" type="String" length="255"
        widget="dropdown" lookup-code="Lookup.Users.Role" required="true"/>
    <AttributeReference name="User Manager" attr-ref="usr_manager_key" available-in-bulk="false"
type="Long" length="382"
        widget="ENTITY" required="false" entity-type="USER"/>
    <AttributeReference name="Country" attr-ref="Country" available-in-bulk="false" type="String"
length="100"
        widget="text" required="false"/>
    <AttributeReference name="Common Name" attr-ref="Common Name" available-in-bulk="false"
type="String" length="240"
        widget="text" required="false" mls="false"/>
    <AttributeReference name="Display Name" attr-ref="Display Name" available-in-bulk="false"
type="String" length="382"
        widget="text" required="false" mls="true"/>
    <AttributeReference name="Department Number" attr-ref="Department Number" available-in-bulk="false"
type="String" length="80"
```

```

        widget="text" required="false"/>
    <AttributeReference name="Description" attr-ref="Description" available-in-bulk="false"
type="String" length="2000"
        widget="text" required="false"/>
    <AttributeReference name="Employee Number" attr-ref="Employee Number" available-in-
bulk="false" type="String" length="80"
        widget="text" required="false"/>
    <AttributeReference name="Fax" attr-ref="Fax" available-in-bulk="false" type="String"
length="20"
        widget="text" required="false"/>
    <AttributeReference name="Generation Qualifier" attr-ref="Generation Qualifier" available-in-
bulk="false" type="String" length="20"
        widget="text" required="false" mls="false"/>
    <AttributeReference name="Home Phone" attr-ref="Home Phone" available-in-bulk="false"
type="String" length="20"
        widget="text" required="false"/>
    <AttributeReference name="Hire Date" attr-ref="Hire Date" available-in-bulk="false"
type="Date" length="50"
        widget="date" required="false"/>
    <AttributeReference name="Home Postal Address" attr-ref="Home Postal Address" available-in-
bulk="false" type="String" length="256"
        widget="text" required="false"/>
    <AttributeReference name="Locality Name" attr-ref="Locality Name" available-in-bulk="false"
type="String" length="80"
        widget="text" required="false"/>
    <AttributeReference name="Email" attr-ref="Email" available-in-bulk="false" type="String"
length="256"
        widget="text" required="false"/>
    <AttributeReference name="Mobile" attr-ref="Mobile" available-in-bulk="false" type="String"
length="20"
        widget="text" required="false"/>
    <AttributeReference name="Pager" attr-ref="Pager" available-in-bulk="false" type="String"
length="20"
        widget="text" required="false"/>
    <AttributeReference name="Postal Address" attr-ref="Postal Address" available-in-
bulk="false" type="String" length="256"
        widget="text" required="false" mls="false"/>
    <AttributeReference name="PO Box" attr-ref="PO Box" available-in-bulk="false" type="String"
length="20"
        widget="text" required="false"/>
    <AttributeReference name="Postal Code" attr-ref="Postal Code" available-in-bulk="false"
type="String" length="30"
        widget="text" required="false"/>
    <AttributeReference name="usr_locale" attr-ref="usr_locale" available-in-bulk="false"
type="String" length="80"
        widget="text" required="false"/>
    <AttributeReference name="State" attr-ref="State" available-in-bulk="false" type="String"
length="80"
        widget="text" required="false" mls="false"/>
    <AttributeReference name="Street" attr-ref="Street" available-in-bulk="false" type="String"
length="80"
        widget="text" required="false"/>
    <AttributeReference name="Telephone Number" attr-ref="Telephone Number" available-in-
bulk="false" type="String" length="20"
        widget="text" required="false"/>
    <AttributeReference name="Title" attr-ref="Title" available-in-bulk="false" type="String"
length="80"
        widget="text" required="false" mls="false"/>
    <AttributeReference name="Initials" attr-ref="Initials" available-in-bulk="false"
type="String" length="10"
        widget="text" required="false"/>

```

```

    <AttributeReference name="Start Date" attr-ref="Start Date" available-in-bulk="false" type="Date"
length="50"
        widget="date" required="false"/>
    <AttributeReference name="End Date" attr-ref="End Date" available-in-bulk="false" type="Date"
length="50"
        widget="date" required="false"/>
    <AttributeReference name="LDAP Organization Unit" attr-ref="LDAP Organization Unit" available-in-
bulk="false" type="String" length="80"
        widget="text" required="false" mls="false"/>
    <AttributeReference name="LDAP Organization" attr-ref="LDAP Organization" available-in-bulk="false"
type="String" length="80"
        widget="text" required="false" mls="false"/>
    <AttributeReference name="usr_timezone" attr-ref="usr_timezone" available-in-bulk="false"
type="String" length="100"
        widget="text" required="false" mls="false"/>

    <AttributeReference name="Number Format" attr-ref="Number Format" available-in-bulk="false"
type="String" length="30"
        widget="dropdown" lookup-code="Lookup.Users.NumberFormat" required="false"
mls="false"/>
    <AttributeReference name="Currency" attr-ref="Currency" available-in-bulk="false" type="String"
length="20"
        widget="dropdown" lookup-code="Lookup.Users.Currency" required="false" mls="false"/>
    <AttributeReference name="Date Format" attr-ref="Date Format" available-in-bulk="false"
type="String" length="20"
        widget="dropdown" lookup-code="Lookup.Users.DateFormat" required="false" mls="false"/>
    <AttributeReference name="Time Format" attr-ref="Time Format" available-in-bulk="false"
type="String" length="20"
        widget="dropdown" lookup-code="Lookup.Users.TimeFormat" required="false" mls="false"/>
    <AttributeReference name="Embedded Help" attr-ref="Embedded Help" available-in-bulk="false"
type="String" length="10"
        widget="dropdown" lookup-code="Lookup.Users.EmbeddedHelp" required="false" mls="false"/>
    <AttributeReference name="Font Size" attr-ref="Font Size" available-in-bulk="false" type="String"
length="10"
        widget="dropdown" lookup-code="Lookup.Users.FontSize" required="false" mls="false"/>
    <AttributeReference name="Color Contrast" attr-ref="Color Contrast" available-in-bulk="false"
type="String" length="10"
        widget="dropdown" lookup-code="Lookup.Users.ColorContrast" required="false"
mls="false"/>
    <AttributeReference name="Accessibility Mode" attr-ref="Accessibility Mode" available-in-
bulk="false" type="String" length="20"
        widget="dropdown" lookup-code="Lookup.Users.AccessibilityMode" required="false"
mls="false"/>
    <AttributeReference name="FA Language" attr-ref="FA Language" available-in-bulk="false"
type="String" length="100"
        widget="text" required="false"/>
    <AttributeReference name="FA Territory" attr-ref="FA Territory" available-in-bulk="false"
type="String" length="100"
        widget="text" required="false"/>
    <AttributeReference name="User Name Preferred Language" attr-ref="User Name Preferred Language"
available-in-bulk="true" type="String" length="20" widget="lookup-query" required="false">
    <lookupQuery lookup-query="select mls_locale_code as USR_NAME_PREFERRED_LANG from mls_locale where
( locale_flag=0 OR locale_flag=1 ) order by mls_locale_code asc" display-
field="USR_NAME_PREFERRED_LANG" save-field="USR_NAME_PREFERRED_LANG"/>
    </AttributeReference>

    <Attribute name="Roles" available-in-bulk="false" type="Clob" length="2048" widget="text"
required="false"/>
    <Attribute name="Policy Name" available-in-bulk="false" type="Clob" length="1024" widget="text"
required="false"/>
    <Attribute name="RequestorID" available-in-bulk="false" type="Clob" length="1024" widget="text"

```

```
required="false"/>
  <Attribute name="FAOpData" available-in-bulk="false" type="Clob" length="4096" widget="text"
required="false" />
</request-data-set>
```

A.1.5 Child Data

You might need attributes to store multiple values or attributes that are made up of other attributes. To do so, you can configure one or more child attributes. For example, an Email ID attribute of an entity type User needs to store multiple values. Therefore, you can configure it in the request dataset in the following way:

```
<Attribute name="Email">
  <Attribute name="ID" length="20" type="string" widget="text" />
</Attribute>
```

You might also require an attribute to be composed of multiple attributes. For example, a Oracle Apps User Responsibilities attribute needs to be made up of three attributes: Responsibility Start Date, Responsibility End Date, and Responsibility Name. You can configure this attribute in the request data set in the following way:

```
<AttributeReference name="Oracle Apps User Responsibilities" attr-
ref="UD_RESPONS" type="String" length="20" widget="text" available-in-
bulk="true">
  <AttributeReference name="Responsibility Start Date" attr-
ref="Responsibility Start Date" type="Date" widget="date" required="false"
available-in-bulk="true" length="100" />
  <AttributeReference name="Responsibility End Date" attr-ref="Responsibility
End Date" type="Date" widget="date" required="false" available-in-bulk="true"
length="100" />
  <AttributeReference name="Responsibility Name" attr-ref="Responsibility
Name" type="String" length="30" widget="lookup" required="false" available-in-
bulk="true" lookup-code="Oracle.Responsibility.Name" primary="true"/>.
</AttributeReference>
```

Here, the association of Responsibility Start Date, Responsibility End Date, and Responsibility Name are maintained and the three attributes together constitute a value of the Oracle Apps User Responsibilities child attribute.

Note:

- Only one level of child attributes are supported in Oracle Identity Manager. Therefore, in the example, the Responsibility Start Date, Responsibility End Date, or Responsibility Name attributes cannot be composed of other attributes. Similarly, attribute references cannot have child attributes.
- The values for AttributeReference name and attr-ref must be the same for child table attributes. For instance, in the Oracle Apps User Responsibilities attribute example in this section, the value for both AttributeReference name and attr-ref is Responsibility Start Date.

During request creation, child data is shown in a table and child data can added from a popup window. In this scenario, the requester might want to add multiple

responsibilities with same start date and end date. You can allow the requester to select multiple responsibilities with same start date and end date by specifying the Responsibility Name as primary.

The primary property allows the requester to select multiple values to the Responsibility Name attribute in the window displayed when the requester tries to add a child row. For the Responsibility Start Date and Responsibility End Date attributes, only single value can be provided. With this multiple rows will be added to the child table one for each responsibility name selected with same value of start date and end date.

The following example shows the sample XML code for a request dataset for provisioning an E-Business resource:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<request-data-set
  xmlns = "http://www.oracle.com/schema/oim/request"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  operation = "PROVISION"
  entity = "eBusiness Suite User"
  name = "ProvisionResourceeeBusiness Suite User"
  xsi:schemaLocation = "http://www.oracle.com/schema/oim/request">
  <AttributeReference
    itresource-type = "eBusiness Suite UM"
    available-in-bulk = "true"
    required = "true"
    length = "20"
    widget = "itresource-lookup"
    type = "Long"
    attr-ref = "EBS Server"
    name = "EBS Server"/>
  <AttributeReference
    available-in-bulk = "true"
    length = "240"
    widget = "text"
    type = "String"
    attr-ref = "Description"
    name = "Description"/>
  <AttributeReference
    available-in-bulk = "false"
    length = "240"
    widget = "text"
    type = "String"
    attr-ref = "Email"
    name = "Email"/>
  <AttributeReference
    available-in-bulk = "true"
    length = "80"
    widget = "text"
    type = "String"
    attr-ref = "Fax"
    name = "Fax"/>
  <AttributeReference
    available-in-bulk = "false"
    length = "256"
    widget = "text"
    type = "String"
    attr-ref = "SSO User ID"
    name = "SSO User ID"/>
  <AttributeReference
    available-in-bulk = "false"
    length = "30"
```

```

        widget = "text"
        type = "String"
        attr-ref = "Person ID"
        name = "Person ID"/>
<AttributeReference
    available-in-bulk = "true"
    length = "10"
    widget = "text"
    type = "String"
    attr-ref = "UD_EBS_RESP"
    name = "eBusiness Suite Responsibilities">

    <AttributeReference
        name = "Application Name"
        attr-ref = "Application Name"
        type = "String"
        length = "256"
        widget = "lookup-query"
        available-in-bulk = "true"
        required = "true">
    <lookupQuery
        lookup-query = "select lkv_encoded as Value,lkv_decoded as
Description from lkv lkv, lku lku where lkv.lku_key=lku.lku_key and
lku_type_string_key='Lookup.EBS.Application' and instr(lkv_encoded,concat('$Form
data.EBS Server', '~'))>0"
        display-field = "Description"
        save-field = "Value"/>
    </AttributeReference>
<AttributeReference
    name = "Responsibility Name"
    attr-ref = "Responsibility Name"
    type = "String"
    length = "256"
    widget = "lookup-query"
    available-in-bulk = "true"
    required = "true"
    primary = "true">
    <lookupQuery
        lookup-query = "select lkv_encoded as Value,lkv_decoded as
Description from lkv lkv,lku lku where lkv.lku_key=lku.lku_key and
lku_type_string_key='Lookup.EBS.Responsibility' and
instr(lkv_encoded,concat('$Form data.Application Name', '~'))>0"
        display-field = "Description"
        save-field = "Value"/>
    </AttributeReference>
<AttributeReference
    available-in-bulk = "true"
    length = "20"
    widget = "date"
    type = "Date"
    attr-ref = "Effective Start Date"
    name = "Effective Start Date"/>
</AttributeReference>
<AttributeReference
    available-in-bulk = "true"
    length = "10"
    widget = "text"
    type = "String"
    attr-ref = "UD_EBS_RLS"
    name = "eBusiness Suite User Role Grants">
    <AttributeReference

```

```

name = "Application Name"
attr-ref = "Application Name"
type = "String"
length = "256"
widget = "lookup-query"
available-in-bulk = "true"
required = "true">
<lookupQuery
  lookup-query = "select lkv_encoded as Value,lkv_decoded as Description
from lkv lkv, lku lku where lkv.lku_key=lku.lku_key and
lku_type_string_key='Lookup.EBS.Application' and instr(lkv_encoded,concat('$Form
data.EBS Server', '~'))>0"
  display-field = "Description"
  save-field = "Value"/>
</AttributeReference>
<AttributeReference
  name = "Role Name"
  attr-ref = "Role Name"
  type = "String"
  length = "256"
  widget = "lookup-query"
  available-in-bulk = "true"
  required = "true"
  primary = "true">
<lookupQuery
  lookup-query = "select lkv_encoded as Value,lkv_decoded as Description
from lkv lkv,lku lku where lkv.lku_key=lku.lku_key and
lku_type_string_key='Lookup.EBS.UMX.Roles' and instr(lkv_encoded,concat('$Form
data.Application Name', '~'))>0"
  display-field = "Description"
  save-field = "Value"/>
</AttributeReference>
<AttributeReference
  available-in-bulk = "true"
  length = "20"
  widget = "date"
  type = "Date"
  attr-ref = "Start Date"
  name = "Start Date"/>
</AttributeReference>
</request-data-set>

```

??

In the sample XML code for provisioning an E-Business resource dataset:

- The Oracle Apps User Responsibilities attribute is defined as a parent attribute for the Responsibility Start Date, Responsibility End Date, and Responsibility Name child attributes. Users can specify one or more values the Oracle Apps User Responsibilities. In request creation UI, this is shown as a table with header "Oracle Apps User Responsibilities" and with the Responsibility Start Date, Responsibility End Date, and Responsibility Name columns.

For the parent attribute, value of attr-ref should be the child table name in the process form. In this example it is "UD_RESPONS".

The Responsibility Start Date, Responsibility End Date, and Responsibility Name attributes are the columns of the child table "UD_RESPONS".

For the child attributes, the attr-ref value must be the Field Label value in the child table of the process form.

Oracle Identity Manager allows you to define a child process form and associates it to a parent process form for a resource. The attributes in the parent form are modeled as attribute references in the request dataset. The attributes in the child form are modeled as attribute references in the child data.

Consider the example of a request based on the Provision Resource request model for a E-Business resource. The following table shows details of the parent process form definition along with child process form details:

Child Form	Attribute Name
UD_RESPONS	Responsibility Start Date
	Responsibility End Date
	Responsibility Name

See above example for the Provision E-Business Resource request dataset.

- For `AttributeReference name="Server"`, the value of `widget` is specified as `itresource-lookup`. This indicates that for the `Server` field, a lookup with available IT resource parameters will be available to the user. If `widget="itresource-lookup"`, then a value for the `itresource-type` element must be specified. For example, `itresource-type="EBIZServer"` indicates that for the `Server` lookup field, all the IT resource parameters for the `EBIZServer` IT resource type must be available for selection. User can search and select an it resource instance by using this lookup.

Note:

IT resource type is a template for all IT resource definitions associated with the connector. An IT resource type specifies the parameters that are common to all IT resource instances, such as host servers and computers, of that particular IT resource type. See *Managing Connector Lifecycle* for more information about IT resources and IT resource types.

See Also:

[Step 6: Creating Request Templates](#) for information about how the attributes are displayed in the Create Request Template wizard

A.1.6 Common Request Dataset

Oracle Identity Manager has a default dataset that is common for all the resources. The common request dataset defines an attribute that is common to all the resources.

`ResourceCommonDataset` is the default common dataset that is common for all the resources. This defines the `Service Account` attribute, which is common across all resources.

As a result, even if the selected resource does not have a dataset, the attribute from the common dataset is shown in request data collection during request creation.

During the request data collection, attributes from both common dataset and the resource specific dataset are shown. In other words, the request collection data is a union of the common dataset and the dataset associated with the entity.



Note:

The common request dataset is shipped with Oracle Identity Manager, and it cannot be customized.

A.1.7 Configuring Localized Values for Request Datasets

All default request datasets have translations for dataset attributes. But for custom request datasets that you create, localization for those dataset attributes must be added to custom resource bundles.

This section describes the configuration of localized values for request datasets and dataset attributes in the following sections:

- [Localization for Request Dataset Attributes](#)
- [Localization of Column Names in LookupQuery for Dataset Attributes](#)

A.1.7.1 Localization for Request Dataset Attributes

This section describes the conventions for enabling localization support for dataset attributes. As an example, the `CreateUserDataSet.xml` defines the role attribute as follows:

```
<AttributeReference name="Role" attr-ref="Role" available-in-bulk="false"
type="String" length="20" widget="dropdown" lookup-code="Lookup.Users.Role"
required="true"/>
```

The translation for this attribute is configured as:

```
request.dataset.User.Role=USER_TYPE
```

Here, `request.dataset.User.Role` is the translation key, and `USER_TYPE` is the actual translation or translation value. Framing the translation key decides the actual translation. Any translation key that is related to request dataset attributes must start with `request.dataset`. This is followed by an object name, such as an entity type, entity subtype if present, parent attribute name if present, attribute name, and predefined value for attribute. The types of translation keys are divided into the following categories:

- [Request Dataset Attributes for Nongeneric Request Models](#)
- [Request Dataset Attributes for Generic Request Models](#)
- [Child Attributes](#)
- [Predefined Values for Attributes](#)

A.1.7.1.1 Request Dataset Attributes for Nongeneric Request Models

The translation key for request datasets of nongeneric request models must not contain an entity subtype because as it is not present in the dataset. For example, the Role attribute in `CreateUserDataSet.xml` is defined as follows:

```
<AttributeReference name="Role" attr-ref="Role" available-in-bulk="false"
type="String" length="20" widget="dropdown" lookup-code="Lookup.Users.Role"
required="true"/>
```

The translation key is:

```
request.dataset.User.Role
```

Here:

- **request.dataset** is the fixed string that must be present at the beginning of any dataset attribute.
- **User** corresponds to the entity type. This must be same as the entity-type property of request-model element in the CreateUserRequestModel.xml file. It can be Resource or Role depending on the request model.
- **Role** corresponds to the actual attribute for which translation is being added. This corresponds to the name property of the Role attribute reference in the dataset.

A.1.7.1.2 Request Dataset Attributes for Generic Request Models

The translation key for request datasets of generic request models must contain entity subtype. For example, for EBS Server attribute in ProvisionResourceBusiness Suite User.xml, the following must be defined:

```
<AttributeReference name="EBS Server" attr-ref="EBS Server" type="Long"
widget="itresource-lookup" required="true" available-in-bulk="true" itresource-
type="eBusiness Suite UM" length="40"/>
```

The translation key is:

```
request.dataset.Resource.eBusiness\ Suite\ User.EBS\ Server=EBS Server
```

Here:

- **request.dataset** is the fixed string that must be present at the beginning of any dataset attribute.
- **Resource** corresponds to the entity type in the ProvisionResourceRequest.xml file.
- **eBusiness\ Suite\ User** is the entity subtype, which is the same as entity property of request-data-set element in ProvisionResourceBusiness Suite User.xml. This is optional and is present in this example because the Provision Resource request model is of generic type.
- **EBS\ Server** corresponds to the actual attribute for which translation is being added. This corresponds to name property of EBS Server attribute reference in the dataset.

A.1.7.1.3 Child Attributes

For child attributes, the translated key additionally contains the parent attribute name that is necessary to uniquely identify the attribute. For example, the Application Name attribute in ProvisionResourceBusiness Suite User.xml, as shown:

```
<AttributeReference available-in-bulk="true" length="10" widget="text"
type="String" attr-ref="UD_EBS_RESP" name="EBS_RSO">
  <AttributeReference name="Application Name" attr-ref="APPLICATION_NAME"
type="String" length="256" widget="lookup-query" available-in-bulk="true"
```

```

required="true">
  <lookupQuery lookup-query="select lkv_encoded, lkv_decoded from lkv lkv, lku
lku where lkv.lku_key=lku.lku_key and lku_type_string_key='Lookup.EBS.Application' and
instr(lkv_encoded,concat('$Form data.EBS Server', '~'))>0" display-field="lkv_decoded"
save-field="lkv_encoded"/>
</AttributeReference>
</AttributeReference>

```

APPLICATION_NAME is a child attribute of EBS_RSO attribute. The translation key for *APPLICATION_NAME* consists of:

- request.dataset.Resource.eBusiness\ Suite\ User.EBS_RSO.Application\ Name = *APPLICATION_NAME*
- **request.dataset** is the fixed string that must be present at the beginning of any dataset attribute.
- **Resource** corresponds to the entity type in the ProvisionResourceRequest.xml file.
- **eBusiness\ Suite\ User** is the entity property of request-data-set element in ProvisionResourceeBusiness Suite User.xml . This is also referred as entity subtype. This is optional and is present in this example because Provision Resource request model is of generic type.
- **EBS_RSO** is the parent attribute of Effective Start Date attribute, which identifies it uniquely. This corresponds to the name property of EBS_RSO attribute reference.
- **Application\ Name** corresponds to the actual attribute for which translation is being added. This corresponds to the name property of *APPLICATION_NAME* attribute reference.

A.1.7.1.4 Predefined Values for Attributes

You can have attributes that have predefined set of values, such as lookup values or lookup-code values. Translations can be added for these values as well because these values are displayed on the user interface. The translation key for this type of value is similar to that of child attributes. For example, the Role attribute in the CreateUserDataSet.xml request dataset has predefined set of values, including Employee, Full-Time Employee, and Part-Time Employee. These values are displayed as drop-down in the UI for Create User Request. These values are predefined by configuring a Lookup Definition with code Lookup.Users.Role. This is specified as lookup-code="Lookup.Users.Role" in the Role AttributeReference, as shown in the following example:

```

<AttributeReference name="Role" attr-ref="Role" available-in-bulk="false"
type="String" length="20" widget="dropdown" lookup-code="Lookup.Users.Role"
required="true"/>

```

Translation for Employee value of Role attribute is:

```
request.dataset.User.Role.LOV.Employee=Employee
```

Here:

- **request.dataset** is the constant part which must be prefixed to every dataset attribute.
- **User** corresponds to the entity type.
- **Role** corresponds to the actual attribute for which Employee is a predefined value. This corresponds to name property of Role AttributeReference in the dataset.
- **LOV** is to be added to specify that the string following LOV is the predefined value for Role.

- **Employee** is the predefined value for which the translation is being added. This must be the value in Decode column of lookup definition.

 **Note:**

The lookup definitions have Code Key and Decode columns. For example, Code Key = EMP, Decode=Employee.

The attributes in the default request datasets are already present. However, if you want to add any new attributes to the dataset, then you can also add translations for these attributes in the same way as described in this section, depending on which category they fall into.

A.1.7.2 Localization of Column Names in LookupQuery for Dataset Attributes

In request datasets, there might be attributes that are of type lookup-query. For example:

```
<AttributeReference name="Application Name" attr-ref="APPLICATION_NAME"
type="String" length="256" widget="lookup-query" available-in-bulk="true"
required="true">
  <lookupQuery lookup-query="select lkv_encoded as Application
Key,lkv_decoded as Application Name from lkv lkv, lku lku where
lkv.lku_key=lku.lku_key and lku_type_string_key='Lookup.EBS.Application'"
display-field="APPLICATION_NAME" save-field="APPLICATION_KEY"/>
</AttributeReference>
```

The columns mentioned in the query are displayed on the UI in a data collection step during request creation. You can localize the lkv_encoded and lkv_decoded column names by adding the column aliases as keys in resource bundle. For example, in the above query, Application Key and Application Name are the aliases for lkv_encoded and lkv_decoded columns respectively. They can be localized by adding translation to custom resource bundles, as follows:

Application\ Key=APPLICATION_KEY

Application\ Name=APPLICATION_NAME

A.2 Step 2: Uploading Request Datasets into MDS

After creating a request dataset XML file, it must be uploaded to MDS, which can be done by using the MDS import/export utility tools provided by Oracle Identity Manager. When the upload is done, request engine loads the dataset during request creation, the attribute references and attributes are shown in data collection step. Similarly, you can delete or export any dataset file from MDS repository by using similar tools.

To upload the request datasets to MDS:

 **Note:**

- The dataset must be updated from time to time based on the resource requirement changes.
- For updating an existing dataset, make sure that there are no pending requests that use that dataset.

1. The `metadata_from_loc` property in the `weblogic.properties` file specifies the top-level directory from which to import XML files. Create a subdirectory to keep the request datasets in, and copy the dataset into this directory. It is recommended that you create a subdirectory structure such as `/custom/RESOURCE_NAME`. For example, if the `metadata_from_loc` property is set to `/scratch/datasets/upload` and you are creating a dataset for the EBS resource, then the dataset is to be placed in the `/scratch/datasets/upload/custom/EBS/` directory.

 **Note:**

Make sure that this directory contains only the required datasets and no other files.

2. Go to the `OIM_HOME/bin` directory and run `weblogicImportMetadata.sh` or `weblogicImportMetadata.bat` script. To do so, set up the environment for the MDS utilities as follows:
 - a. Set the `OIM_ORACLE_HOME` environment variable to the Oracle Identity Management Oracle home directory inside the Middleware home directory. For example, for Microsoft Windows, set the `OIM_ORACLE_HOME` environment variable to `C:\Oracle\Middleware\Oracle_IDM1\` directory.
 - b. Set the necessary properties in the `weblogic.properties` file, which is located in the same folder as the utilities.

Table A-3 Parameters in the Properties File

Property Name	Description	Notes
<code>wls_servername</code>	Name of the Oracle WebLogic Server on which Oracle Identity Manager is deployed	
<code>application_name</code>	The application name	Value is: <ul style="list-style-type: none"> • <code>oim</code> if importing/exporting an out-of-the-box event handler. • <code>OIMMetadata</code> for customizable metadata. If importing or exporting custom data, set <code>application_name</code> to <code>OIMMetadata</code> .

Table A-3 (Cont.) Parameters in the Properties File

Property Name	Description	Notes
metadata_from_loc	Directory location from which an XML file should be imported. This property is used by <code>weblogicImportMetadata.sh</code> script.	Microsoft Windows paths include <code>//</code> as file or directory separator.
metadata_to_loc	Directory location to which an XML file should be exported. This property is used by the <code>weblogicExportMetadata.sh</code> script.	Microsoft Windows paths include <code>//</code> as file or directory separator.
metadata_files	Full path and name of an XML file. This property is used by <code>weblogicExportMetadata.sh</code> and <code>weblogicDeleteMetadata.sh</code> scripts.	For example, you may specify <code>/file/User.xml</code> to export a user entity definition. You can indicate multiple xml files as comma-separated values.

- **Using the Import utility:** When you run the `weblogicImportMetadata.sh` utility, all files specified in `metadata_from_loc` will be imported.

For example, you want to import `User.xml` which exists in `/scratch/johnny/temp/oim/file/User.xml`. You must define `metadata_from_loc` as `/scratch/johnny/temp/oim`.

 **WARNING:**

Make sure no other files exist in the directory specified by `metadata_from_loc` or its subdirectories. The import utility tries to recursively import all the files under the directory.

All the Import, Export, and Delete utilities are located in the `OIM_ORACLE_HOME/server/bin/` directory.

- **Using the Export utility:** When you run the `weblogicExportMetadata.sh` utility, the files specified in `metadata_files` will be exported to the folder specified by `metadata_to_loc`.

 **Note:**

If the file name contains spaces, then you must specify the file name as is, without quotes or any escape characters. For example, to export a file named `ProvisionResourceeBusiness Suite User.xml`, where the path is `/db/ProvisionResourceeBusiness Suite User.xml`, you must specify the `metadata_files` property as follows:

```
metadata_files=/db/ProvisionResourceeBusiness Suite User.xml
```

- **Using the Delete utility:** When you run the `weblogicDeleteMetadata.sh` utility, the files specified in `metadata_files` will be deleted from MDS.

 **Note:**

If the file name contains spaces, then you must specify the file name as is, without quotes or any escape characters. For example, to delete a file named ProvisionResourceeBusiness Suite User.xml, where the path is /db/ProvisionResourceeBusiness Suite User.xml, you must specify the metadata files property as follows:

```
metadata_files=/db/ProvisionResourceeBusiness Suite User.xml
```

A.3 Step 3: Creating SOA Composites Required for Approval

The primary goal of any provisioning system is to manage requests submitted by users and provision resources to users. Request completion involves execution of associated approval processes. These approval processes are deployed as Service Oriented Architecture (SOA) composites running on the SOA Server. Request service is responsible for execution and management of such approval processes.

This section contains the following topics:

- [Using SOA Composites for Approval](#)
- [Creating a SOA Composite](#)

A.3.1 Using SOA Composites for Approval

The interaction between Oracle Identity Manager and SOA Server is explained in the following steps:

1. The user creates a request by using the Oracle Identity Manager Self Service. The request can be of any one of all the request types supported by Oracle Identity Manager.
2. Request service evaluates the approval policy, and the SOA composite to be instantiated is selected.

 **Note:**

The composites must be registered with Oracle Identity Manager in order for them to be kicked off when the request is submitted. For information about registering workflows with Oracle Identity Manager, see "[Step 4: Registering the SOA Composites in Oracle Identity Manager](#)".

3. Request service contacts SOA Server to instantiate the selected SOA composite. Instantiates selected composite instance on SOA server.
4. SOA composite run starts and human approval task is assigned for approval.
5. The approver logs in to the Task List in the Oracle Identity Manager Self Service console, and approves the request.
6. After the approval, the composite instance run is completed, which is notified to the request service.
7. Request service moves the request to the next stage.

Oracle Identity Manager provides a few predefined SOA composites. However, you can define your own composites and use them in request approvals.

To use a SOA composite as an approval process, it must adhere to certain standards. These standards ensure that the request service is able to instantiate and manage such composites correctly. These standards are:

- The following attributes are mandatory for BPEL process:
 - RequestID of type String
 - RequestModel of type String
 - RequestTarget of type String
 - URL of type String
 - RequesterDetails of XML Element
 - BeneficiaryDetails of XML Element
 - ObjectDetails of XML Element
 - OtherDetails of XML Element

The RequestID, RequestModel, RequestTarget, and URL attributes are always set with valid values for all types of requests.

RequesterDetails is an XML element. This element is filled up with valid values for all requests that requires authentication. Requester details is empty for the requests of type Self-Register User because the requester is anonymous user.

BeneficiaryDetails is an XML element. This element is filled up with valid values for all requests that have a beneficiary, for example, Provision Resource and Assign Roles. This is filled up only if the request is associated with single beneficiary. If the request is associated with multiple beneficiaries, then BeneficiaryDetails is empty. BeneficiaryDetails element always has valid value for simple requests and child requests that have a beneficiary. Therefore, it is recommended to use this XML element in SOA composites that are used as approval processes at the operational level of approval. This is because at the operational level of approval, the request is associated with only one beneficiary.

ObjectDetails is an XML element. This element is filled up with valid values for all requests that are associated with the Resource entity. This is filled up only if the request is associated with single resource. If the request is associated with multiple resources, then ObjectDetails is empty. The ObjectDetails element always has valid value for simple and child requests that are associated with resource. Therefore, it is recommended to use this XML element in SOA composites that are used as approval processes at the operational level of approval. This is because at the operational level of approval, the request is associated with only one resource.

- All the attributes that are mandatory for the BPEL process are referred from RequestDetails.xsd and ApprovalProcess.xsd. These files are present in the template SOA composite, which must not be modified or deleted.

A.3.2 Creating a SOA Composite

Oracle Identity Manager provides a helper utility for creating custom SOA composites. This utility creates a template SOA project that adheres to all the necessary standards. This utility is located in the *OIM_HOME/workflows/new-workflow* directory.

 **Note:**

- JAVA_HOME environment variable must be set before running this utility.
- This utility requires Apache Ant version 1.7 or later.

To create a custom SOA composite by running the helper utility:

1. Run the following commands:

```
cd OIM_HOME/workflows/new-workflow
ant -f new_project.xml
```

2. Enter the JDeveloper application name when the following prompt is displayed:

```
Please enter application name
```

3. Enter the JDeveloper project name when the following prompt is displayed:

```
Please enter project name
```

4. Enter the name of the ADF binding service for the composite when the following prompt is displayed:

```
Please enter the service name for the composite. This needs to be unique
across applications
```

The new application is created in the `OIM_HOME/workflows/new-workflow/process-template/` directory. You can open the new application in JDeveloper for modification.

Human task in the template SOA composite is configured to send notifications to the assignee of the human task. In the custom composite that is created, the notification message can be modified based on the requirement. All the notifications to be sent to the approver must be configured in the SOA composite. For configuring Oracle SOA server to send notifications, see *Installing and Configuring SOA and User Messaging Service in Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

Human task in the template SOA composite is configured to be assigned to the SYSTEM ADMINISTRATORS role.

A.4 Step 4: Registering the SOA Composites in Oracle Identity Manager

Note:

If the communication to Oracle Identity Manager is through the SSL mode, then you must perform the following prerequisites for communication to Oracle Identity Manager through SSL mode:

- Set the *TRUSTSTORE_LOCATION* environment variable, where *TRUSTSTORE_LOCATION* is the trusted key store file location.
- Use t3s protocol instead of t3. For example, the URL for Oracle Identity Manager is:

```
t3s://HOST_NAME:PORT
```

The SOA composite must be registered with Oracle Identity Manager before it can be used as an approval process. To register a SOA composite with Oracle Identity Manager:

1. Create the *COMPOSITE_NAME.props* property file in the *OIM_HOME/workflows/registration/* directory with the following content:

```
name=COMPOSITE_NAME
category=Approval
providerType=BPEL
serviceName=REQUEST_APPROVAL_SERVICE
domainName=DOMAIN
version=REVISION_ID
payloadID=PAYLOAD
operationID=OPERATION_ID
listOfTasks=HUMAN_TASK_NAMES
```

Where:

- Replace *COMPOSITE_NAME* with the name of the SOA composite as specified in the *composite.xml* file.
- Replace *REQUEST_APPROVAL_SERVICE* with the name of the service exposed in the composite. This service is invoked while instantiating the composite for approval.
- Replace *OPERATION_ID* with the name of operation to be invoked on the service mentioned for the *serviceName* property.
- Replace *PAYLOAD* with the part name of the operation specified as the value of the *operationID* property.
- Replace *REVISION_ID* with the SOA composite revision as specified in the *composite.xml* file.
- Replace *DOMAIN* with name of SOA partition in which the composite is deployed. By default, SOA has one partition named default.

- Replace *HUMAN_TASK_NAMES* with the names of the approval tasks associated with the SOA composite. Separate the names with colon (:).

 **Note:**

Do not add any extra spaces in the properties file.

2. Run the following command from the *OIM_HOME/workflows/registration/* directory:

 **Note:**

- *JAVA_HOME* environment variable must be set before running this utility.
- This utility requires Apache Ant version 1.7 or later.

```
ant -f registerworkflows-mp.xml register
```

3. Enter Oracle Identity Manager administrator username when prompted.
4. Enter Oracle Identity Manager administrator password when prompted.

 **Note:**

When you login to the SOA server by using the Oracle Identity Manager administrator credentials, the login attempt fails if your challenge questions and answers and password are not reset in Oracle Identity Manager. You must reset the challenge questions and answers and password when you first login to Oracle Identity Manager.

5. Enter Oracle Identity Manager server t3 URL when prompted.
6. Enter the complete path of the property file name that you created in Step 1 when prompted.

A.5 Step 5: Defining Request Approvals

A request goes through multiple approvals before it is executed. After the request is submitted, it must obtain approvals at different levels. Approvals are controlled and configured by a set of approval policies.

An approver is able to view the request data. Approver cannot change the data provided by the requester. Approver can only provide data for the attributes that are set as `approver-only="true"` in the request dataset.

This section describes the following topics:

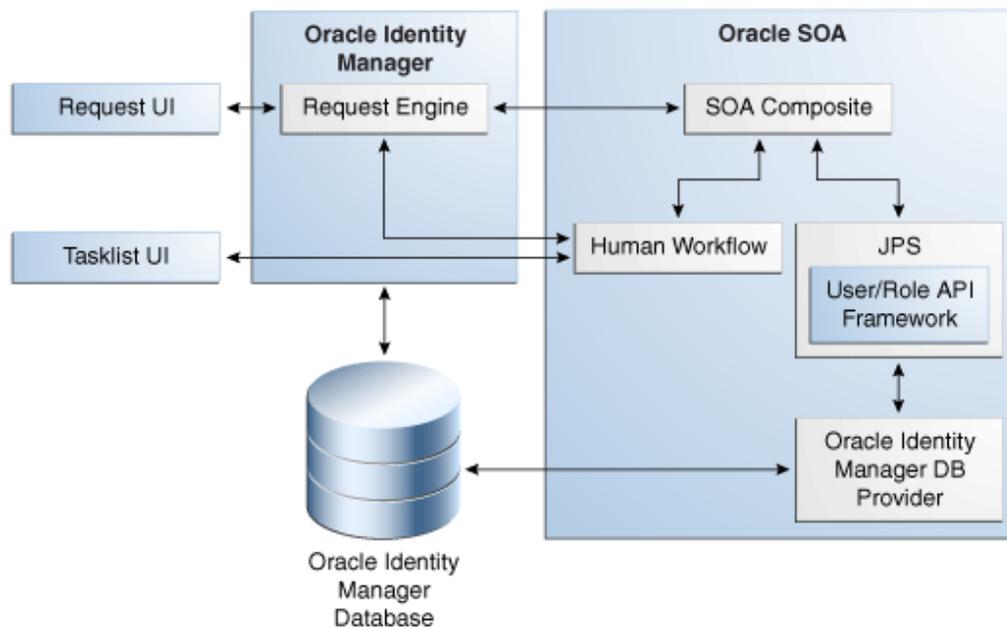
- [Approval Workflows](#)
- [Approval Levels](#)
- [Creating Approval Policies](#)

A.5.1 Approval Workflows

After a request is submitted, if any approval needs to be initiated, then the request service starts the workflow process in the backend workflow engine. Oracle SOA is used as workflow engine by the request service.

SOA server hosts SOA composites and human workflow. The integration of the request service and SOA can be explained with the help of [Figure A-1](#):

Figure A-1 Request Service and SOA Integration



The following process describes how Oracle SOA works with request service for selecting an approval workflow:

1. A request is created by using the request management UI, which is Oracle Identity Manager Self Service or Advanced Administration.
2. When the request is submitted, the request engine calls the SOA composites that are deployed in Oracle SOA.

 **Note:**

Oracle SOA is independent of Oracle identity Manager. The backend Business Process Execution Language (BPEL) service invokes the approval workflow. In addition to the default BPEL workflows shipped with Oracle Identity Manager, you can define your own workflows in BPEL based on your requirement. For information about customizing the BPEL workflows, see Getting Started with Oracle BPEL Process Manager in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

3. Oracle SOA determines whom to assign the request by using the SOA composite logic through Java Platform Security (JPS). Oracle SOA uses the same set of users and roles as in Oracle Identity Manager. This is enabled by the Oracle Identity Manager database provider.
4. Oracle SOA assigns the task to the assignee based on the information provided by the DB provider.
5. The list of assigned tasks to the logged in user and role is displayed in the TaskList UI.
6. Using the TaskList, the approver approves or rejects the requests.
7. The approval outcome is send back to the request engine through SOA.
8. If a request has been approved, then the next action is determined based on the request type, beneficiary, or associated resource. If a request has been rejected, then the request processing stops.

After the task is assigned, the user can login to the TaskList UI in Oracle Identity Manager Self Service to get a consolidated view of Human tasks and Oracle Identity Manager requests.

TaskList uses the task query service APIs to communicate with Oracle SOA. These APIs are provided by Oracle SOA server. Oracle Identity Manager uses the SOAP or RMI protocol to communicate with Oracle SOA based on the configuration. RMI is the default protocol.

A.5.2 Approval Levels

Each request may need to go through three levels of approvals: template-level, request-level, and operation-level. SOA composites must be registered with Oracle Identity Manager in order to use those as approval processes in Oracle Identity Manager. Registering means letting Oracle Identity Manager know what approval processes are deployed and can be used at run time.

The approval levels are described in the following sections:

- [Template-Level Approval](#)
- [Request-Level Approval](#)
- [Operation-Level Approval](#)

A.5.2.1 Template-Level Approval

These are the approvals defined at the request templates. Each template can define additional approvals on top of what is defined by the approval policy configuration. At this level, either the complete request is approved or rejected. For bulk requests, there is no partial approval or rejection. Each template can define an optional approval process that must be initiated at the template level. If no approval process is defined in the template, then the template level is auto approved.

Note:

Request-level and operation-level approval are associated with approval policies. For template-level approval, there are no approval policies associated. The approval processes are defined at the template and are run directly, without an association with approval policies.

An example of template-level approval can be the approval required by an HR representative for the user creation of all contract employees in addition to the approval required from the employees' managers and IT administrators. The additional approval from the HR representative can be configured as an approval process while creating a template. The template can be used to create and submit the request.

**Note:**

This level of approval is not required for child requests.

A.5.2.2 Request-Level Approval

These are the approval for the entire request. These are based on the approval policy configuration.

An example of request-level approval is the approval required by the requestor's manager when a request is raised to provision a laptop to a user.

Approval process that needs to be used for the request-level approval for a request is determined by the approval policies defined at the request level. If no approval policies are defined for the request level for the given type of request, then the default request approval process is used. By default, all the request-level approval are assigned to the administrator. Therefore, the default configuration is secure. If multiple approval policies exist, then the approval policy rules are evaluated in the order of approval policy priority to figure out the appropriate approval policy. The approval policy rules indicate the request engine which approval process to pick up for the particular approval. The request engine selects the approval process defined in the approval policy that is selected based on the approval policy rule evaluation.

For example, when a request to create a user is submitted, the approval policy selection methodology finds out how many approval policies exist for the Create User request model in the order of priority. The approval policy rule for the highest approval policy priority is evaluated. An example of approval policy rule is that manager's first name and last name must be John and Doe respectively. If the approval policy rule for the highest priority does not match, then the approval policy rule for the next approval policy priority is evaluated. When the first approval policy rule matches the criteria specified in the approval policy, the corresponding approval process for that approval policy is selected for that request at the request level. If all the approval policy rules do not match, then the default approval process for the request level is selected.

**Note:**

- Request-level approval is not required for child requests because the bulk request is broken down to child requests after successful approval of the bulk request at the request level.
- The Self-Register User request template has organization as the approver-only attribute. Therefore, any approval policy associated with this template must not have auto-approval set at request-level because the attribute is mandatory and must be provided by the approver.

A.5.2.3 Operation-Level Approval

These are the approvals for the operation being performed by this request type. This level requires approval selection methodology name and parameters to be passed to the methodology. A methodology suggests which approval workflow is to be used for this operation. The request type and scope may also define the methodology-specific parameters that are required for determining the approval process. Scope is a key associated with the types of request types, as shown in [Table A-4](#):

Table A-4 Request Types and Associated keys

Request Types	Scope
All request types related to the user entity, such as:	organization
<ul style="list-style-type: none"> • Create User • Self-Register User • Modify Self Profile • Modify User Profile • Delete user • Enable User • Disable User 	
All request models related to resources, such as:	resource
<ul style="list-style-type: none"> • Provision Resource • Modify Provisioned Resource • Self-Request Resource • Enable Provisioned Resource • Disable Provisioned Resource • Modify Provisioned Resource • Self Modify Provisioned Resource • De-provision Resource • Self De-Provision Resource 	
All request types related to the role entity, such as:	role
<ul style="list-style-type: none"> • Assign Roles • Remove from Roles 	



Note:

For the Create Role, Modify Role, and Delete Role types, the operation-level approval is auto-approved.

For example, based on the scope, for the request of type Provision Resource, you must select the resource to associate the approval policy at operation level at the time of approval policy creation. Similarly, for a Create User request type, you must select an organization, and for the Assign Roles request type, you must select a role at the operation level during approval policy creation.

The approval policies that you create, along with the approval policy priorities, approval policy rules, and scope, decide which approval process is to be selected for a request at the operation level.

An example of operation-level approval is the approval required by the IT administrator, who is responsible for issuing a laptop to users, after the request-level approval is obtained for provisioning a laptop to a user.

For a bulk request, operation-level approval are required for individual child requests. Each individual child request can be approved or rejected independently. For example, for a provision resource to user request, there can be multiple beneficiaries, multiple resources, or both. Therefore, at the operational level, provisioning of each resource to each user generates a child request, which can be approved or rejected independently.

A.5.3 Creating Approval Policies

Create the required approval policies for selecting appropriate SOA composites for approval. See "[Step 5: Defining Request Approvals](#)" for the concepts related to approvals.

To create an approval policy:

1. In Oracle Identity Manager Advanced Administration, click the **Policies** tab, and then click **Approval Policies**. Alternatively, you can click **Search Approval Policies** under Policies in the Welcome page.
2. From the Actions menu on the left pane, select **Create**. You can also start the Create Approval Policy wizard by clicking the icon with the plus (+) sign on the toolbar. The Step 1. Set Approval Policy Details page of the Create Approval Policy wizard is displayed.
3. Enter values for the following fields, and then click **Next**:
 - **Policy Name**: Enter a name for the approval policy. This is a mandatory attribute.
 - **Description**: Enter the details about what this approval policy will do.
 - **Request Type**: Select the request type by selecting from the LOV, for example, Assign Roles. This is a mandatory attribute.
 - **Level**: Select the approval level that you want to implement for this approval policy. This is a mandatory attribute.
 - **Scope Type**: Set automatically based on the request type selection. For example:
 - If request type is set to Create User, then Scope Type is automatically set to Organization.
 - If request type is set to Provision Resource, then Scope Type is automatically set to Resource.
 - If request type is set to Assign Roles, then Scope Type is automatically set to Role.

 **Note:**

The Scope Type, All Scope, and Scope fields are applicable only if the Level field is set to Operation Level. These fields are disabled if the Level field is set to Request Level.

- **All Scope:** Select this option to specify the approval policy associated with all entities for a particular entity type. For example, for the Create User request type, Scope Type is Organization. If you select All Scope, then this approval policy is applicable to all organizations in Oracle Identity Manager. This is same for Resource and Role scopes.
 - **Scope:** Select this option to specify the approval policy associated with the specific entity for a particular entity type. An approval policy can be associated with a specific Scope based on the Scope Type.

The Scope field is disabled if All Scope field is set. If All Scope field is not set, then this field becomes mandatory and must be set with some value.
 - **Auto Approval:** Select this option to specify automatic approval at the request level or operation level that you select in the Level field.
 - **Approval Process:** Select the workflow that you want to associate with this approval policy. If Auto Approval is selected, then this field is disabled and you cannot set any value. If Auto Approval is not set, then this field becomes mandatory.
4. On the Step 2. Set Approval Rule and Component page, enter the name of the approval policy rule in the Rule Name field, for example, RuleTest1.
 5. In the Rule Components section, you can define the parameters of the approval policy rule. To do so, click the icon next to the View list. The Add Simple Rule dialog box is displayed. In this dialog box, you must select values for the following fields, and then click **Save**.
 - **Entity:** Entity, such as Requester, Beneficiary, or Resource, with which the approval policy rule is associated. This varies based on the selected request type and the approval level.
 - **Attribute:** Attribute of the above selected entity.
 - **Condition:** Condition of the approval policy rule, such as Equals, Not Equals, or Starts With.
 - **Value:** Value of the condition.

 **Note:**

If you use the User Login attribute in a rule expression, the corresponding User Login ID value must be entered in all uppercase letters, otherwise the expression will not evaluate to true.

- **Parent Rule Container:** The rule container with which this approval policy rule needs to be associated with.

 **Note:**

When writing simple rule expressions, if an entity attribute has an encoded value, then create the expression by using the encoded value, not the lookup-code definition. For example, for the account status attribute, create the expression by using the encoded value 1 or 0, not the decoded value Locked or Unlocked.

6. Rule containers can be used for modeling complex conditions with And and Or combinations. To add a rule container for the approval policy rule, in the Rule Components section, from the Actions menu, select **Add Rule Container**. The Add Rule Container dialog box is displayed. In this dialog box, enter or select values for the following fields, and then click **Add**.
 - **Rule Container Name:** The name of the rule container.
 - **Parent Rule Container:** The name of the rule container under which you want to create this rule container. A rule container can hold either another rule container or rule elements with the AND or OR operators in a hierarchical order.
 - **Operator:** The operators are AND and OR.
7. After the approval rule creation is complete, click **Next**.
8. On the Step 3. Review Approval Policy Summary page, verify the information that you have specified for the approval policy. You can click the Back button to modify any information if you want. Click **Finish** to create the approval policy.
9. A message is displayed confirming that the approval policy has been created. Click **OK**.

A.6 Step 6: Creating Request Templates

This section contains the following topics:

- [About Request Templates](#)
- [Creating a Request Template Based on the Create User Request Type](#)
- [Creating a Request Template Based on the Provisioning Resource Request Type](#)

A.6.1 About Request Templates

A request template lets you customize a request type for a purpose. In other words, it allows you to control the attributes of the request by controlling the various capabilities in the UI. For instance, if you want to create requests for user creation for all contract employees and specify an attribute to have a particular value, then you can customize the Create User request type to create a request template that allows customization of the request. By creating the request template, you can specify that the organization for all employees must be XYZ Inc. or the user type of all contract employees must be Part-time Employee.

Access to templates for request creation is based on the role assignment defined in the template. After creation of a request template, it is available only to the users with the roles that are assigned to the template.

A default template is shipped predefined for each of the request type. These predefined templates can not be deleted or renamed. Names of these predefined templates is same as corresponding models.

You can use a request template for the following purposes:

- **Adding template-level approval:** You can add an additional level of approval apart from request-level and operation-level while creating the template.
- **Adding restrictions:** This includes:
 - **Adding entity restrictions:** You can specify restrictions of the entity types that can be selected through the request templates. For example, a template for Provisioning Resource request type might specify a number of valid resources that can be selected by using this template. This limits the use of the template to specific type of entities in case of generic requests. For example, the template defined on provisioning request type may specify that this template can only be used for Active Directory, Exchange, and UNIX resources.

 **Note:**

If no entity type is restricted in the template, then all the available entity types are shown to the requester while creating the request by using this template.

However, the data to be collected during various phases of the request lifecycle is controlled by request datasets.

- **Restricting data values for an attribute:** If you specify a value for attributes, then the default value of the attribute is set, and the attribute is not displayed in the UI. On specifying multiple such values, the values are available to the user as List of Values (LOV), from which the user can select a value.

Attribute restriction can be of the following types:

- * Specifying a default value to an attribute in the request template. During the request creation using this template, this attribute is not shown to the requester. This attribute and the corresponding value is set automatically in the request data.
 - * Restricting an attribute with multiple values in request template. On specifying multiple such values, the values are available to the requester as List of Values (LOV), from which the requester can select a value during the request creation by using this template.
 - * Restricting an attribute with no value in request template, by selecting the **Do not allow users to enter values for this attribute** option. This type of restriction is allowed only for the nonmandatory attributes. With this restriction, during the request creation by using this template, this attribute is not shown to the requester. This attribute will not be part of request data.
- **Adding additional data collection attributes:** These attributes are not associated with any entity. Data collected by using such mechanism cannot be used during request execution. However, it can be used for reporting purpose, validations on the request, and postsubmission data action handlers.

You can define new attributes in a request template that are shown to the requester during request creation in the additional data collection step. These attributes are specific to this template and are not associated with any entity.

- **Assigning roles to template to restrict the use by end users:** Only the members of the appropriate roles assigned to the template can create a request by using that template.

To summarize, the following are achieved by using the request template:

- The restricted entity types can be specified.
- The restricted attributes that are not required to be collected as a part of the request for the entity can be specified.
- The attribute can be restricted to one value or list of values. If only one value is specified, then the attribute is not shown to the requester while submitting the request. If a list of values is specified, then the requester has to select one value from the list of values.
- Additional data collection attributes can be specified.
- Roles can be assigned to templates to restrict the use by end users.

The template management service internally uses Oracle Entitlements Server (OES) for determining who can perform what operations. The OES policy for request template authorization specifies that only users with the REQUEST TEMPLATE ADMINISTRATORS role are authorized to create or clone, search, modify, and delete request templates.

As a user belonging to the REQUEST TEMPLATE ADMINISTRATORS role, you can create a request template by using the Create Request Template wizard in the UI for request management. Steps in the wizard are dynamically generated based on the selection of the request type in the first step and the selection of resource for resource-based request types.

Creation of request templates is described with the help of the following scenarios:

- [Creating a Request Template Based on the Create User Request Type](#)
- [Creating a Request Template Based on the Provisioning Resource Request Type](#)

A.6.2 Creating a Request Template Based on the Create User Request Type

To create a request template based on the Create User request type:

1. Log in to Oracle Identity Manager Administrative and User Console with credentials that have the permission to create a request template.

 **Note:**

The user who is a member of the REQUEST TEMPLATE ADMINISTRATORS role is allowed to create a request template. If the appropriate role is not assigned to the user, then the required UI options for creating a request template will not be available to the user.

2. Click **Advanced** to open Oracle Identity Manager Advanced Administration.

3. Click the **Configuration** tab, and then click **Request Templates**. Alternatively, you click the **Search Request Templates** link under Configuration in the Welcome page.
4. On the left pane, from the Actions menu, select **Create**. Alternatively, you can click the **Create Request Template** icon on the toolbar. The Set request template details page of the Create Request Template wizard is displayed.
5. Enter values for the following fields, and then click **Next**.
 - **Request Template Name:** Enter the name of the template that you want to create, for example, Create Contractor.
 - **Request Type:** Select the type of request for which you want to create the request template, for example, Create User.
 - **Description:** Enter a description for the request template that you are creating.
 - **Template Level Approval Process:** Specify the approval workflow name if you want to specify an approval process for the Create User request. This is a template-level approval in addition to the request-level and operation-level approvals. For creating users for contract employees, you can specify that the HR representative, who is responsible for the recruitment of all contract employees, must approve the user creation.

Figure A-2 shows the Set request template details page of the Create Request Template wizard:

Figure A-2 The Set Request Template Details Page

6. On the Select Attributes to Restrict page, select the attributes of the Create User type for which you want the user to enter values. Attributes that are restricted by the request templates are either not shown to the user, or the user is only allowed to select from predefined LOVs. User cannot enter any values. Figure A-3 shows the Select Attributes to Restrict page:

Figure A-3 The Select Attributes to Restrict Page

The screenshot shows a web application window titled 'Create Request Template'. The window has a breadcrumb trail at the top: 'Step 1: Details', 'Step 2: Attributes' (highlighted), 'Step 3: Restrictions', 'Step 4: Additional Attributes', 'Step 5: User Roles', and 'Step 6: Confirm'. Below the breadcrumb, the page title is 'Select Attributes to Restrict'. On the right side, there is a legend '* Indicates required fields.' and three buttons: 'Cancel', 'Back', and 'Next'. The main content area is titled 'User Attributes' and contains a list of attributes, each with a checkbox:

- First Name
- Middle Name
- Last Name
- User Login
- Password
- Organization
- Design Console Access
- User Type
- User Manager
- Country
- Common Name
- Display Name
- Department Number
- Description
- Employee Number

This page displays the attributes based on the dataset for Create User request type. If a request is created by using the Create User request template, then you can specify values for all these default attributes. If you want to restrict some of these attributes and want the requester to enter values for a few attributes, then you can select those attributes in this page. For example, you can select Middle Name because a value for this attribute must be specified. In this example, you can select the Middle Name, Organization, User Type, User Manager, and Country attributes.

 **Note:**

- Even if a dataset attribute is configured with a PrePopulationAdapter, it can be restricted in a request template. In such case, pre-population will not happen and the values restricted in template will be shown in Request creation UI. Hence, if pre-population is required for an attribute, it should not be restricted in the template.
- As mentioned earlier in this section, the steps in the wizard are dynamically generated based on the request type and the resource selection for resource-based request types. The steps are indicated on the top of the tab.

7. On the Set Attribute Restrictions page, specify restrictions on the attributes that you selected in the Select Attributes to Restrict page. To specify restrictions:

 **Note:**

This step is generated only if there are any attributes specified in the corresponding request data set.

- a. For the User Login attribute, select any one of the following:

- **Do not allow users to enter values for this attribute:** Select this option if you do not want the user to specify a value for the attribute. On selecting this option, the attribute will not be displayed in the UI when creating the user. This option is not displayed for a mandatory attribute because a value must be specified for a mandatory attribute.

- **Restrict this attribute to the following values:** Select this option if you want to specify one or more values for the attribute. For example, if you specify a value for the Department Number attribute, such as Software Engineering, then the default value of the attribute is set to Software Engineering, and the attribute is not displayed in the UI when creating a request by using this template. You can also specify multiple values for the attribute by using the + (plus) icon. On specifying multiple values, the values are available to the user as LOVs when creating a request by using this template, from which the user can select a value.

 **Tip:**

These options are displayed for the Department Number attribute because the attribute is specified as a text box in the request dataset.

- b. Specify one or more values for the Organization attribute. To do so, click the search icon next to the Organization field, select one or more organization names from the Available Organizations list, and clicking the **Move** button.

 **Tip:**

The Organization attribute is displayed as a field for which you must select a value by searching the existing organization names because this attribute is specified as an entity in the request dataset. This is a dynamic LOV because organizations can be created in Oracle Identity Manager.

- c. Specify a value for the User Type attribute. To do so, select one or more values from the Available User Type list, and click the **Move** button.

 **Tip:**

The User Type attribute is displayed as a static LOV because this attribute is specified as a static LOV in the request dataset. This is a static LOV because the user must select from the available user types and cannot create new user types.

- d. Specify values for the User Manager and Country attributes, and click **Next**.

Figure A-4 shows the Set Attribute Restrictions page:

Figure A-4 The Set Attribute Restrictions Page

Note:

Steps 5, 6, and 7 are common for all request templates creation.

8. On the Set Additional Attributes page, you can specify additional information about attributes, which need to be collected based on the template that you are creating but are not used for the purpose of entity mapping.

Note:

The Additional Attribute Data is not used during request execution. This data is also not displayed to the approver.

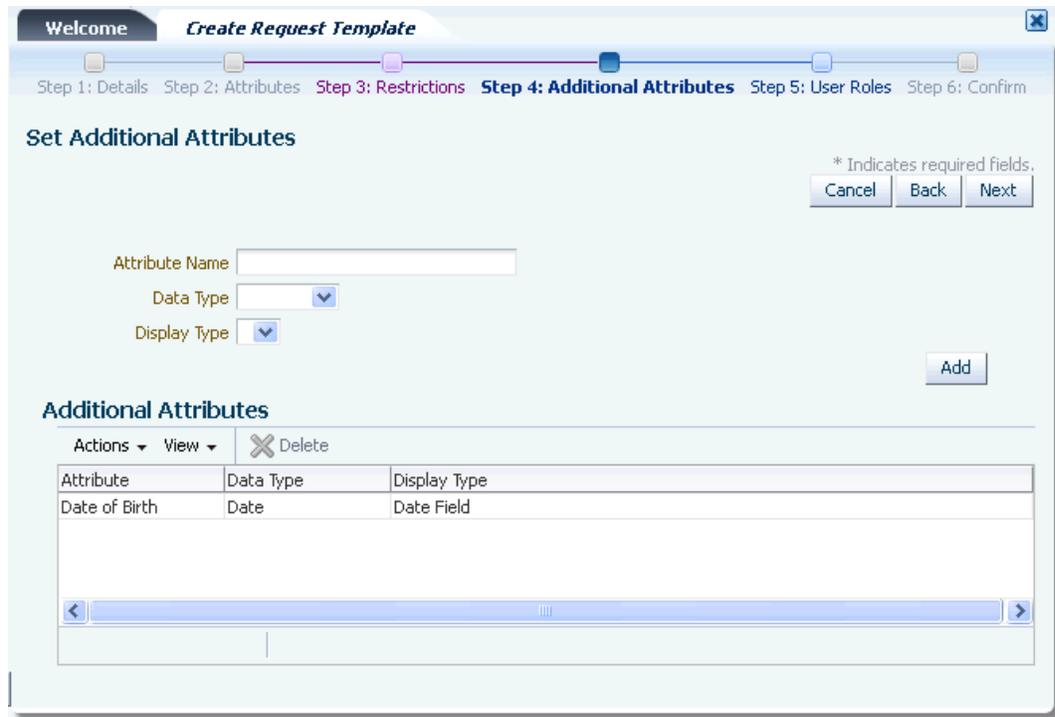
In this example, specify date of birth as the additional attribute name. Select the Data Type as **Number** and Display Type as **Text Field**, and then click **Add**. You can specify multiple attributes by clicking the **Add** button. When finished, click **Next**.

See Also:

"[Step 1: Creating a Request Dataset for the Resources](#)" for more information about the additional attributes that are not mapped to the underlying Oracle Identity Manager entity

Figure A-5 shows the Set Additional Attributes page:

Figure A-5 The Set Additional Attributes Page



9. On the Set Template User Roles page, you can select one or more roles, for example, AD Administrators, whose members are allowed to create requests by using the template that is being created. In this example, from the Available Roles list, select a role such as **Contractor Administrators**. Click **Move** to include the selected roles in the Selected Roles list, and then click **Next**.

Each request template can be associated with a set of roles. Only the users with any of these roles are able to create a request by using this template. When a new request template is created with a list of associated roles, a new authorization policy is created internally. In addition, if the role association with any of the existing request templates is modified (adding new roles or removing existing roles), then the existing authorization policy for this template is modified.

The default authorization policy for creating requests by using request template allows users with the REQUEST TEMPLATES ADMINISTRATORS role to access all operations related to request templates. The policy has the following details:

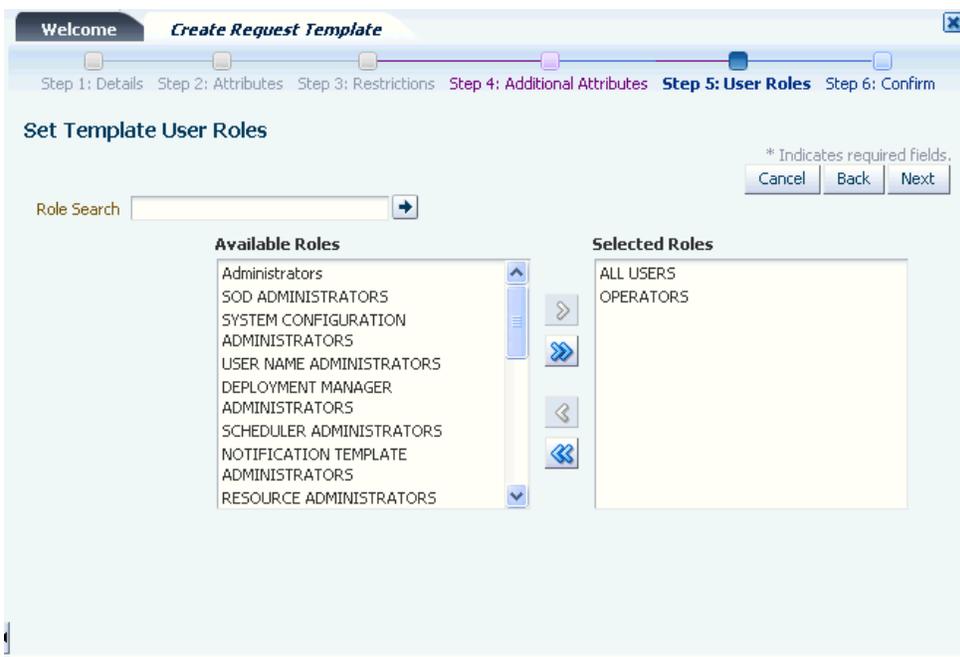
- **Policy name:** Request Template Administration Policy
- **Assignee:** REQUEST TEMPLATE ADMINISTRATORS role
- **Functional security:** The permissions are:
 - Create
 - Delete
 - Modify
 - Search

These permissions do not support fine-grained attribute-level controls.

- **Data security:** None
- **Description:** Allows users with the REQUEST TEMPLATE ADMINISTRATORS or SYSTEM ADMINISTRATORS role to access all request template actions.

Figure A-6 shows the Set Template User Roles page:

Figure A-6 The Set Template User Roles Page



10. On the Review Request Template Summary page, as shows in Figure A-7, review the data that have been entered for Request Template Name, Request Type, Description, and Template Level Approval Process, and then click **Finish**.

Figure A-7 The Review Request Template Summary Page

Welcome **Create Request Template**

Step 1: Details Step 2: Attributes Step 3: Restrictions Step 4: Additional Attributes Step 5: User Roles Step 6: Confirm

Review Request Template Summary:

* Indicates required fields.

Cancel Back Finish

Request template basic information

Request Template Name	Create User
Request Type	Create User
Description	
Template Level Approval Process	default/BeneficiaryManagerApproval!1.0

11. Click **OK** to confirm the template creation.

In the Create Request Template wizard, the following steps are common irrespective of the request type that you select or the request dataset that you define:

- Request details to be specified in the Set request template details page. See step 5 in the create request templates.
- Setting additional attributes in the Set Additional Attributes page. See step 8.
- Setting roles for the template in the Set Template User Roles page. See step 9.
- Request template information in the Review Request Template Summary page. See step 10.

A.6.3 Creating a Request Template Based on the Provisioning Resource Request Type

The Provision Resource default request template that is based on the Provision Resource request type can be used for provisioning resources to users. But if you want to customize the request creation for provisioning specific resources to users, then you can create a request template, which is based on the Provision Resource request type.

To create a request template based on the Provisioning Resource request type:

1. In Oracle Identity Manager Advanced Administration, click the **Configuration** tab, and then click the **Request Templates** tab. Alternatively, you click the **Search Request Templates** link under Configuration in the Welcome page.

 **Note:**

The user who is a member of the REQUEST TEMPLATE ADMINISTRATORS role is allowed to create a request template. If the appropriate role is not assigned to the user, then the required UI options for creating a request template will not be available to the user.

2. On the left pane, from the Actions menu, select **Create**. Alternatively, you can click the **Create a Request Template** icon on the toolbar. The Set request template details page of the Create Request Template wizard is displayed.
3. Enter values for the following fields, and then click **Next**.
 - **Request Template Name:** Enter the name of the request template, for example, Provision E-Business Resource.
 - **Request Type:** Select a request type, such as Provision Resource.

 **Note:**

The steps in the Create Request Template wizard are dynamically generated on clicking **Next** after providing the Request Template Basic Information in the first step of the wizard.

- **Description:** Enter a description for the request template that you are creating.
 - **Approval Process:** Enter the name of the approval workflow. For information about this field, see Step 4 of "[Creating a Request Template Based on the Create User Request Type](#)".
4. In the Select Allowed Resources page, click **Search** to search for all the available resources.
 5. From the Available Resources list, select one or more resources, and then click **Move** to include the selected resources in the Selected Resources list. In this example, select the **E-Business RO** resource, and then click **Next**.

 **Note:**

- Only the resources that you select in this step are displayed to the requester during request creation by using this template. If you do not select a resource here, then all the resources in Oracle Identity Manager are displayed while creating the request.
- If no entity type is restricted in the template, then all the available entity types are shown to the requester while creating request using this template.

6. In the Select Attributes to Restrict page, select the attributes associated with the E-Business resource that you want to restrict. These attributes are defined in the request dataset for provisioning the E-Business resource. See "[Step 1: Creating a Request Dataset for the Resources](#)" for more information about attributes.

If you select multiple resources in the Select Allowed Resources page, then the attributes associated with all the resources are displayed in the Select Attributes to Restrict page. Select the attributes for all the resources that you want to restrict, and then click **Next**.

7. In the Set Attribute Restrictions page, specify values for the attributes whose values you want to restrict. For example, for the Fax attribute, select the **Do not allow users to enter values for this attribute** option if you do not want the user to specify a value for the attribute. Otherwise, select the **Restrict the attribute to the following values** option and specify one or more values for the Fax attribute. For information about these options and setting restrictions for attributes, see "[Creating a Request Template Based on the Create User Request Type](#)".

Note that the **Do not allow users to enter values for this attribute** option is not available for the Server and Life Span Type attributes. This is because these attributes are specified as required in the request dataset. For information about the required property, see "[Creating a Request Template Based on the Create User Request Type](#)".

Select restriction values for all the attributes, and then click **Next**.

 **Tip:**

If you are creating a request template for a request to provision multiple resources to users, click the **Next Resource** and **Previous Resource** buttons to set attribute restrictions for all the resources.

 **Note:**

Attributes coming up as shuttle on attribute restrictions page will show upto 200 results at a time. You need to provide appropriate search pattern to get relevant search results.

8. Perform steps 8 through 10 of the procedure in "[Creating a Request Template Based on the Create User Request Type](#)" to complete the wizard.

 **Note:**

In the Create Request Template wizard, the steps to select resources and set attribute restrictions vary based on the request type. The rest of the steps are similar.

While creating a request template, if you select a resource that does not have a request dataset defined, then you are not allowed to restrict the attributes to collect from the user. This is because there is no information specified about the data that is to be collected from the user for the selected resource. As a result, the Step 3: Attributes and Step 4: Restrictions in the Create Request Template wizard are not applicable because the attributes in these steps are defined by the request dataset, in the absence of which, there is no data to restrict. However, when you select a resource that does not have a request dataset, the Service Account attribute is displayed in the Step 3: Attributes because this attribute is defined by the common request dataset. See "[Common Request Dataset](#)" for information about common request dataset.

A.7 Extending Request Management Operations

You can customize certain aspects of request management operations to allow greater flexibility and implement customized logic for additional functionality. To achieve this, you can use request management plug-ins. There are plug-in points that you can use to implement customization.

This section discusses the plug-in points in the following topics:

- [Running Custom Code Based on Request Status Change](#)
- [Validating Request Data](#)
- [Prepopulation of an Attribute Value During Request Creation](#)

A.7.1 Running Custom Code Based on Request Status Change

In Oracle Identity Manager, a request undergoes change in status at each stage of its lifecycle. The request engine exposes a plug-in point that allows running of custom code during request status change. A plug-in with custom code that extends this plug-in point can be implemented and registered for running the code. The plug-in point is the **oracle.iam.request.plugins.StatusChangeEvent** interface with the **public void followUpActions(String reqId)** method. This method consists of the request id parameter, using which the request details can be obtained with the help of request management APIs.

Any code that is to be run during the status change must be implemented in the `followUpActions()` method in a plug-in class that implements the `oracle.iam.request.plugins.StatusChangeEvent` interface. You must specify at which request status change this plug-in is to be run in the `plugin.xml` file.

For example, when a request in Oracle Identity Manager moves to the Request Failed status, you want to run a custom code that sends a notification to an administrator. To do so:

1. Create a new plug-in class with name `RequestFailedChangeEvent` that implements the `oracle.iam.request.plugins.StatusChangeEvent` interface. This class must have the logic of sending a notification to the administrator in the `followUpActions(String reqId)` method.
2. Define `plugin.xml` in following standard format, as specified by the plug-in framework:

```
<oimplugins xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <plugins pluginpoint="oracle.iam.request.plugins.StatusChangeEvent">
    <plugin pluginclass="com.mycompany.RequestFailedChangeEvent"
      version="1.0" name="RequestFailedChangeEvent">
      <metadata name="status">
        <value>Request Failed</value>
      </metadata>
    </plugin>
  </plugins>
</oimplugins>
```

In this XML definition, the metadata part specifies at which stage the plug-in must be run. This is done by specifying the metadata value as `Request Failed`, which means that the `com.mycompany.RequestFailedChangeEvent` plug-in will run when a request moves to the Request Failed status.

3. Register the plug-in with Oracle Identity Manager using one of the following ways:

- Using the PlatformService.registerPlugin and PlatformService.unregisterPlugin APIs for registration-related tasks:

Here is an example:

```
ClientPlatform platform = OIMClient.getInstance();
platform.login("username", "password");
PlatformService service = platform.getService(PlatformService.class);
File zipFile = new File(fileName);
FileInputStream fis = new FileInputStream(zipFile);
int size = (int) zipFile.length();
byte[] b = new byte[size];
int bytesRead = fis.read(b, 0, size);
while (bytesRead < size) {
    bytesRead += fis.read(b, bytesRead, size - bytesRead);
}
fis.close();
service.registerPlugin(b);
service.unregisterPlugin(pluginID, version);
```

- Using the Plugin Registration Utility: The utility uses the pluginregistration.xml and ant.properties files. These files are located in the *OIM_HOME/plugin_utility/* directory.

 **Note:**

Plug-in registration utilities require Apache Ant version 1.7 or later.

To register the plug-in using the utility, perform the following steps:

- a. Set the values for *WLS_HOME* and *OIM_HOME* in ant.properties. For example:

```
WLS_HOME = ../middleware/wlserver_10.3
OIM_HOME = ../middleware/Oracle_IDM1/server
```

- b. Build the wfullclient.jar in Oracle WebLogic server by change directories to *WLS_HOME/server/lib* and running the following command:

```
java -jar ../../modules/com.bea.core.jarbuilder_1.3.0.0.jar
```

 **Note:**

The exact JAR file version can be different based on the WLS. Use the corresponding file with the name as *com.bea.core.jarbuilder* at the *WLS_HOME/./modules/* directory.

- c. Execute the ant target "register":

```
ant -f pluginregistration.xml register
```

This will prompt for the Oracle Identity Manager username and password along with the server information and the location of the plugin zip file.

- d. Enter the complete path of the zip file location.

A.7.2 Validating Request Data

You can use the RequestDataValidator plug-in to add custom validation of request data after submission. The plug-in point for this is the **oracle.iam.request.plugins.RequestDataValidator** interface with public void validate(RequestData requesterData) method. See "[The DataSetValidator Element](#)" for information about the DataSetValidator element.

A.7.3 Prepopulation of an Attribute Value During Request Creation

Prepopulation plug-in is associated with an attribute reference or attribute in request dataset. This can be used to prepopulate an attribute value by running custom code during request creation. Requester can modify the value that is prepopulated if required.

The plug-in point for this is **oracle.iam.request.plugins.PrePopulationAdapter** with public Serializable prepopulate(RequestData requestData) method. Use this plug-in only for the following request types:

Provision Resource, Self-Request Resource, Create User, Self-Register User.

See "[The PrePopulationAdapter Element](#)" for more details.

Index