# Contents

## Preface
- Audience
- Documentation Accessibility
- Related Documents
- Documentation Updates
- Conventions

## What's New in the Oracle Identity Manager Connector for Database User Management?
- Software Updates
- Documentation-Specific Updates

## 1 About the Connector
1.1 Certified Components
1.2 Usage Recommendation
1.3 Certified Languages
1.4 Connector Architecture
1.5 Features of the Connector
  1.5.1 Mapping Standard and Custom Attributes for Reconciliation and Provisioning
  1.5.2 Predefined and Custom Reconciliation Queries
  1.5.3 Predefined and Custom Provisioning Queries
  1.5.4 Support for Creating Global and External Users In Oracle Database
  1.5.5 Full and Incremental Reconciliation
  1.5.6 Limited (Filtered) Reconciliation
  1.5.7 Batched Reconciliation
  1.5.8 Specifying Accounts to Be Excluded from Reconciliation and Provisioning Operations
  1.5.9 Connection Pooling
  1.5.10 Support for Connector Server
  1.5.11 Support for Creating Connector Copies
  1.5.12 Transformation and Validation of Account Data
  1.5.13 Support for Reconciling Data About Deleted Entities
  1.5.14 Separate Scheduled Jobs for Reconciliation of Users, Logins, and Deleted Login Entities
1.5.15 Support for SSL Communication Between the Target System and Oracle Identity Manager 1-11
1.5.16 Support for Managing Authorization to Oracle Database Vault Realms................. 1-12
1.5.17 Support for Configuring the Connector for Enterprise User Security...................... 1-12
1.6 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 .................................................................. 2-3
2.2 Installation .................................................................................................................... 2-3
2.2.1 Installing the Connector in Oracle Identity Manager................................................. 2-4
2.2.2 Deploying the Connector Bundle in a Connector Server ......................................... 2-6
2.2.2.1 Installing and Configuring the Connector Server ............................................... 2-6
2.2.2.2 Running the Connector Server ............................................................................ 2-8
2.2.2.3 Installing the Connector on the Connector Server ............................................. 2-8
2.3 Postinstallation ............................................................................................................. 2-9
2.3.1 Postinstallation on Oracle Identity Manager ......................................................... 2-9
2.3.1.1 Configuring the Target System As a Trusted Source ...................................... 2-9
2.3.1.2 Changing to the Required Input Locale ......................................................... 2-10
2.3.1.3 Clearing Content Related to Connector Resource Bundles from the Server Cache ... 2-11
2.3.1.4 Creating the Administrator Account on Oracle Database Vault ...................... 2-12
2.3.1.5 Setting up the Lookup Definition for Connection Pooling ................................ 2-12
2.3.1.6 Enabling Logging for Oracle Identity Manager ................................................ 2-13
2.3.1.7 Configuring Oracle Identity Manager Release 11.1.2 or Later ............................ 2-15
2.3.1.8 Localizing Field Labels in UI Forms .................................................................... 2-18
2.3.2 Configuring the IT Resource for the Target System ................................................. 2-20
2.3.3 Configuring the Connector to Support Multiple Versions of the Target System... 2-22
2.3.4 Configuring the IT Resource for the Connector Server ........................................... 2-24
2.3.5 Enabling Logging for Connector Server .................................................................. 2-26
2.4 Upgrading the Connector .......................................................................................... 2-27
2.4.1 Preupgrade Steps .................................................................................................. 2-27
2.4.2 Upgrade Steps ....................................................................................................... 2-27
2.4.3 Postupgrade Steps ................................................................................................ 2-28
2.5 Postcloning Steps ...................................................................................................... 2-30
2.5.1 Postcloning Configuration for User Accounts ...................................................... 2-31

3 Using and Extending the Connector for MSSQL
3.1 Configuring Secure Communication Between MSSQL and Oracle Identity Manager ... 3-1
3.2 Determining Values for the JDBC URL and Connection Properties Parameters for MSSQL 3-2
3.3 Lookup Definitions for MSSQL .................................................................................. 3-4
3.3.1 Lookup Definitions Synchronized with MSSQL .................................................... 3-4
3.3.2 Lookup Definitions for Configurations for MSSQL ................................................. 3-6
3.3.2.1 Lookup.DBUM.MSSQL.Configuration .............................................................. 3-6
3.3.2.2 Lookup.DBUM.MSSQL_Login.Configuration .................................................. 3-7
3.3.2.3 Lookup.DBUM.MSSQL.UM.Configuration .......................................................... 3-7
3.3.2.4 Lookup.DBUM.MSSQL.UM.Configuration.Trusted .......................................... 3-7
3.3.2.5 Lookup.DBUM.MSSQL.UM.Configuration.Trusted .......................................... 3-8
3.3.3 Lookup Definitions for Attribute Mappings for MSSQL Login Entity ................ 3-8
3.3.3.1 Lookup.DBUM.MSSQL.Login.ProvAttrMap ..................................................... 3-8
3.3.3.2 Lookup.DBUM.MSSQL.Login.ReconAttrMap ................................................... 3-9
3.3.3.3 Lookup.DBUM.MSSQL.Login.ReconDefaults ................................................... 3-9
3.3.4 Lookup Definitions for Attribute Mappings for MSSQL User Entity ................. 3-9
3.3.4.1 Lookup.DBUM.MSSQL.UM.ProvAttrMap ..................................................... 3-9
3.3.4.2 Lookup.DBUM.MSSQL.UM.ReconAttrMap ................................................... 3-10
3.3.4.3 Lookup.DBUM.MSSQL.UM.ReconDefaults ................................................... 3-10
3.3.4.4 Lookup.DBUM.MSSQL.UM.ReconDefaults.Trusted ...................................... 3-10
3.3.4.5 Lookup.DBUM.MSSQL.UM.ReconAttrMap.Trusted ....................................... 3-11
3.3.5 Lookup Definitions for Exclusion Lists for MSSQL ........................................... 3-11
3.3.6 Lookup Definitions for Transformation of Data in MSSQL ............................... 3-12
3.3.7 Lookup Definition for Validation of Data in MSSQL .......................................... 3-12
3.4 Scheduled Jobs for MSSQL .............................................................. 3-12
3.4.1 Scheduled Jobs for Lookup Field Synchronization for MSSQL ....................... 3-12
3.4.2 Attributes for Scheduled Jobs for MSSQL ..................................................... 3-13
3.4.3 Configuring Scheduled Jobs for MSSQL ....................................................... 3-15
3.5 Reconciliation from MSSQL .............................................................. 3-16
3.5.1 Guidelines on Configuring Reconciliation from MSSQL ................................. 3-17
3.5.2 Reconciliation Process for MSSQL .............................................................. 3-17
3.5.3 Reconciliation Queries for MSSQL .............................................................. 3-18
3.5.4 Target System Columns Used in Reconciliation from MSSQL ....................... 3-19
3.5.5 Configuring the Target System As a Trusted Source ....................................... 3-19
3.5.6 Reconciliation Rules for MSSQL ................................................................. 3-20
3.5.7 Reconciliation Action Rules for MSSQL ....................................................... 3-21
3.5.8 Performing Full Reconciliation from MSSQL ................................................ 3-22
3.5.9 Performing Limited Reconciliation from MSSQL ........................................... 3-22
3.5.10 Performing Incremental Reconciliation from MSSQL .................................... 3-22
3.6 Provisioning for MSSQL .............................................................. 3-23
3.6.1 Guidelines on Performing Provisioning Operations for MSSQL ...................... 3-23
3.6.2 Provisioning Process for MSSQL ............................................................... 3-24
3.6.3 Configuring Direct Provisioning for MSSQL .................................................. 3-26
3.6.4 Configuring Request-Based Provisioning for MSSQL ................................. 3-27
3.6.4.1 Approver’s Role in Request-Based Provisioning for MSSQL ....................... 3-27
3.6.4.2 Importing MSSQL Request Datasets Using Deployment Manager ............... 3-28
3.6.4.3 End User’s Role in Request-Based Provisioning for MSSQL ....................... 3-28
3.6.4.4 Enabling the Auto Save Form Feature for MSSQL .................................... 3-29
3.6.4.5 Running the PurgeCache Utility for MSSQL ............................................. 3-29
3.6.5 Switching Between Request-Based Provisioning and Direct Provisioning for MSSQL .... 3-30
3.6.6 Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x 3-31
3.7 Extending the Connector for MSSQL ...................................................... 3-31
3.7.1 Guidelines on Configuring the Queries for MSSQL ........................................ 3-32
3.7.1.1 Syntax of Provisioning Queries for MSSQL ............................................... 3-33
3.7.1.2 Syntax of Reconciliation Queries for MSSQL ............................................. 3-35
3.7.1.3 Syntax of List of Values Queries for MSSQL ............................................. 3-37
3.7.1.4 Guidelines for Configuring Search Queries Used in Reconciliation from MSSQL .... 3-37
3.7.2 Configuring Queries to Add Support for Custom Parameters and Lookup Fields for MSSQL 3-38
3.7.2.1 Updating a Query File for MSSQL ......................................................... 3-38
3.7.2.2 Configuring Oracle Identity Manager ....................................................... 3-40
3.7.3 Configuring the Connector for Multiple Installations of MSSQL ..................... 3-42
3.7.4 Configuring the Connector for Multiple Trusted Source Reconciliation from MSSQL .... 3-45
3.7.5 Configuring Validation of Data During Reconciliation and Provisioning for MSSQL ..... 3-45
3.7.6 Configuring Transformation of Data During User Reconciliation for MSSQL ...... 3-47
3.7.7 Configuring Resource Exclusion Lists for MSSQL ....................................... 3-49
3.7.8 Configuring Action Scripts for MSSQL ....................................................... 3-51

4 Using and Extending the Connector for Oracle Database

4.1 Configuring Secure Communication Between Oracle Database and Oracle Identity Manager 4-1
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 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 ......................... 4-8
4.3.2.1 Lookup.DBUM.Oracle.Configuration .................................................... 4-9
4.3.2.2 Lookup.DBUM.Oracle.UM.Configuration ........................................... 4-9
4.3.2.3 Lookup.DBUM.Oracle.Configuration.Trusted ....................................... 4-10
4.3.2.4 Lookup.DBUM.Oracle.UM.Configuration.Trusted ................................... 4-10
4.3.3 Lookup Definitions for Attribute Mappings for Oracle Database .................... 4-10
4.3.3.1 Lookup.DBUM.Oracle.UM.ProvAttrMap ............................................. 4-11
4.3.3.2 Lookup.DBUM.Oracle.UM.ReconAttrMap .......................................... 4-11
4.3.3.3 Lookup.DBUM.Oracle.UM.ReconAttrMap.Trusted ................................ 4-12
4.3.3.4 Lookup.DBUM.Oracle.UM.ReconDefaults.Trusted .................................. 4-12
4.3.4 Lookup Definitions for Exclusion Lists for Oracle Database .......................... 4-13
4.3.5 Lookup Definitions for Transformation of Data in Oracle Database ................. 4-13
4.3.6 Lookup Definition for Validation of Data in Oracle Database ......................... 4-14
4.4 Scheduled Jobs for Oracle Database ............................................................... 4-14
4.4.1 Scheduled Jobs for Lookup Field Synchronization for Oracle Database ............ 4-14
4.4.2 Attributes for Scheduled Jobs for Oracle Database ..................................... 4-15
4.4.3 Configuring Scheduled Jobs for Oracle Database ....................................... 4-17
4.5 Reconciliation from Oracle Database ............................................................ 4-18
4.5.1 Guidelines on Configuring Reconciliation for Oracle Database ..................... 4-18
| 4.5.2 | Reconciliation Process for Oracle Database .................................................................. 4-19 |
| 4.5.3 | Reconciliation Queries for Oracle Database ............................................................... 4-20 |
| 4.5.4 | Target System Columns Used in Reconciliation from Oracle Database.......................... 4-21 |
| 4.5.5 | Configuring the Target System As a Trusted Source .................................................... 4-21 |
| 4.5.6 | Reconciliation Rules for Oracle Database .................................................................... 4-22 |
| 4.5.7 | Reconciliation Action Rules for Oracle Database ........................................................ 4-23 |
| 4.5.8 | Performing Full Reconciliation from Oracle Database .................................................. 4-23 |
| 4.5.9 | Performing Limited Reconciliation from Oracle Database ............................................. 4-24 |
| 4.5.10 | Performing Batched Reconciliation from Oracle Database ............................................ 4-25 |
| 4.5.11 | Performing Incremental Reconciliation from Oracle Database ...................................... 4-25 |
| 4.6 | Provisioning for Oracle Database .................................................................................. 4-25 |
| 4.6.1 | Guidelines on Performing Provisioning Operations for Oracle Database........................ 4-26 |
| 4.6.2 | Provisioning Process for Oracle Database .................................................................... 4-27 |
| 4.6.3 | Configuring Direct Provisioning for Oracle Database ..................................................... 4-29 |
| 4.6.4 | Configuring Request-Based Provisioning for Oracle Database ...................................... 4-30 |
| 4.6.4.1 | Approver’s Role in Request-Based Provisioning ............................................................ 4-31 |
| 4.6.4.2 | Importing Request Datasets Using Deployment Manager .............................................. 4-31 |
| 4.6.4.3 | End User’s Role in Request-Based Provisioning ............................................................. 4-32 |
| 4.6.4.4 | Enabling the Auto Save Form Feature ........................................................................ 4-32 |
| 4.6.4.5 | Running the PurgeCache Utility .................................................................................. 4-33 |
| 4.6.5 | Switching Between Request-Based Provisioning and Direct Provisioning for Oracle Database | 4-33 |
| 4.6.6 | Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x ............. 4-34 |
| 4.7 | Extending the Connector for Oracle Database ............................................................... 4-34 |
| 4.7.1 | Guidelines on Configuring the Queries for Oracle Database ......................................... 4-35 |
| 4.7.1.1 | Syntax of Provisioning Queries for Oracle Database .................................................... 4-36 |
| 4.7.1.2 | Syntax of Reconciliation Queries for Oracle Database ................................................ 4-38 |
| 4.7.1.3 | Syntax of List of Values Queries for Oracle Database .................................................. 4-40 |
| 4.7.1.4 | Guidelines for Configuring Search Queries Used in Reconciliation from Oracle Database | 4-40 |
| 4.7.2 | Configuring Queries to Add Support for Custom Parameters and Lookup Fields for Oracle Database | 4-41 |
| 4.7.2.1 | Updating the Query Files for Oracle Database ............................................................. 4-41 |
| 4.7.2.2 | Configuring Oracle Identity Manager .......................................................................... 4-43 |
| 4.7.3 | Configuring the Connector for Multiple Installations of Oracle Database ....................... 4-45 |
| 4.7.4 | Configuring the Connector for Multiple Trusted Source Reconciliation from Oracle Database | 4-49 |
| 4.7.5 | Configuring Validation of Data During Reconciliation and Provisioning for Oracle Database | 4-49 |
| 4.7.6 | Configuring Transformation of Data During User Reconciliation for Oracle Database ...... 4-51 |
| 4.7.7 | Configuring Resource Exclusion Lists for Oracle Database .......................................... 4-53 |
| 4.7.8 | Configuring Action Scripts for Oracle Database .......................................................... 4-54 |

5 Using and Extending the Connector for MySQL

5.1 Configuring Secure Communication Between MySQL and Oracle Identity Manager .... 5-1
5.2 Determining Values for the JDBC URL and Connection Properties Parameters for MySQL . 5-2
6 Using and Extending the Connector for DB2

6.1 Configuring Secure Communication Between DB2 and Oracle Identity Manager ........ 6-1
6.2 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 ....................................................... 6-5
6.3.2.3 Lookup.DBUM.DB2.Configuration.Trusted .................................................. 6-5
6.3.2.4 Lookup.DBUM.DB2.UM.Configuration.Trusted ........................................... 6-6
6.3.3 Lookup Definitions for Attribute Mappings for DB2........................................... 6-6
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 .................................................. 6-8
6.3.5 Lookup Definitions for Transformation of Data in DB2....................................... 6-9
6.3.6 Lookup Definition for Validation of Data in DB2................................................ 6-9
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........................................... 6-13
6.5.2 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 .................................... 6-15
6.5.5 Reconciliation Rules for DB2............................................................................... 6-16
6.5.6 Reconciliation Action Rules for DB2................................................................. 6-17
6.5.7 Performing Full Reconciliation from DB2......................................................... 6-18
6.5.8 Performing Limited Reconciliation from DB2.................................................... 6-18
6.5.9 Performing Batched Reconciliation from DB2.................................................. 6-18
6.6 Provisioning for DB2 ......................................................................................... 6-19
6.6.1 Guidelines on Performing Provisioning Operations for DB2 ........................... 6-19
7 Using and Extending the Connector for Sybase

7.1 Configuring Secure Communication Between Sybase and Oracle Identity Manager

7.2 Determining Values for the JDBC URL and Connection Properties Parameters for Sybase

7.3 Lookup Definitions for Sybase

7.3.1 Lookup Definitions Synchronized with Sybase

7.3.2 Lookup Definitions for Configurations for Sybase

7.3.2.1 Lookup.DBUM.Sybase.Configuration

7.3.2.2 Lookup.DBUM.Sybase.Login.Configuration

7.3.2.3 Lookup.DBUM.Sybase.UM.Configuration

7.3.2.4 Lookup.DBUM.Sybase.Configuration.Trusted

7.3.2.5 Lookup.DBUM.Sybase.UM.Configuration.Trusted

7.3.3 Lookup Definitions for Attribute Mappings for Sybase Login Entity

7.3.3.1 Lookup.DBUM.Sybase.Login.ProvAttrMap

7.3.3.2 Lookup.DBUM.Sybase.Login.ReconAttrMap

7.3.3.3 Lookup.DBUM.Sybase.Login.ReconDefaults

7.3.4 Lookup Definitions for Attribute Mappings for Sybase User Entity

7.3.4.1 Lookup.DBUM.Sybase.UM.ProvAttrMap

7.3.4.2 Lookup.DBUM.Sybase.UM.ReconAttrMap

7.3.4.3 Lookup.DBUM.Sybase.UM.ReconDefaults

7.3.4.4 Lookup.DBUM.Sybase.UM.ReconDefaults.Trusted
7.3.4.5  Lookup DBUM Sybase UM Recon AttrMap Trusted ................................................. 7-9
7.3.5  Lookup Definition for Validation of Data in Sybase ...................................................... 7-9
7.3.6  Lookup Definitions for Transformation of Data in Sybase ........................................ 7-10
7.3.7  Lookup Definitions for Exclusion Lists for Sybase ..................................................... 7-10
7.4  Scheduled Jobs for Sybase ................................................................................................. 7-10
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 ..................................................................... 7-13
7.5  Reconciliation from Sybase ............................................................................................. 7-14
7.5.1  Reconciliation Process for Sybase .............................................................................. 7-15
7.5.2  Reconciliation Rules for Sybase ................................................................................ 7-16
7.5.3  Reconciliation Action Rules for Sybase ................................................................... 7-16
7.5.4  Performing Full Reconciliation from Sybase .............................................................. 7-17
7.6  Provisioning for Sybase .................................................................................................... 7-17
7.6.1  Guidelines on Performing Provisioning Operations for Sybase ................................ 7-17
7.6.2  Provisioning Process for Sybase ................................................................................ 7-18
7.6.3  Configuring Direct Provisioning for Sybase ............................................................... 7-20
7.6.4  Configuring Request-Based Provisioning for Sybase ................................................. 7-21
7.6.4.1  Approver’s Role in Request-Based Provisioning .................................................... 7-21
7.6.4.2  Importing Request Datasets Using Deployment Manager .................................... 7-22
7.6.4.3  End User’s Role in Request-Based Provisioning ................................................... 7-22
7.6.4.4  Enabling the Auto Save Form Feature ................................................................ 7-23
7.6.4.5  Running the PurgeCache Utility .......................................................................... 7-23
7.6.5  Switching Between Request-Based Provisioning and Direct Provisioning for Sybase 7-23
7.6.6  Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x 7-24
7.7  Extending the Connector for Sybase ................................................................................ 7-25
7.7.1  Guidelines on Configuring the Queries for Sybase ..................................................... 7-26
7.7.1.1  Syntax of Provisioning Queries for Sybase Database ............................................. 7-26
7.7.1.2  Syntax of Reconciliation Queries for Sybase Database ........................................ 7-28
7.7.1.3  Syntax of List of Values Queries for Sybase Database .......................................... 7-30
7.7.2  Configuring Queries to Add Support for Custom Parameters and Lookup Fields for Sybase 7-30
7.7.2.1  Updating the Query Files for Sybase Database ....................................................... 7-30
7.7.2.2  Configuring Oracle Identity Manager ................................................................... 7-33
7.7.3  Configuring the Connector for Multiple Installations of Sybase ............................... 7-34
7.7.4  Configuring the Connector for Multiple Trusted Source Reconciliation from Sybase 7-38
7.7.5  Configuring Validation of Data During Reconciliation and Provisioning for Sybase 7-38
7.7.6  Configuring Transformation of Data During User Reconciliation for Sybase ....... 7-40
7.7.7  Configuring Resource Exclusion Lists for Sybase ..................................................... 7-42
7.7.8  Configuring Action Scripts for Sybase ...................................................................... 7-44

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 ............................................................................................. 8-6
8.6 Configuring the Resource Object .......................................................................................... 8-8
8.7 Adding Process Tasks, Assigning Adapters, and Mapping Adapter Variables ................ 8-9
8.8 Adding Attributes for Reconciliation ..................................................................................... 8-12
8.9 Configuring Lookup Definitions Used During Connector Operations ............................ 8-12
8.10 Configuring Scheduled Jobs ............................................................................................... 8-14
  8.10.1 Configuring Scheduled Jobs for Lookup Field Synchronization ......................... 8-15
  8.10.2 Configuring Scheduled Jobs for Reconciliation ...................................................... 8-15
8.11 Configuring Oracle Identity Manager for Request-Based Provisioning ....................... 8-16
8.12 Testing the Customized Connector .................................................................................... 8-17

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 .................................................................................. 9-7
  9.5 Running the Test Utility for Sybase .............................................................................. 9-9

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 Updateds 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

Index
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1–1</td>
<td>Architecture of the Connector</td>
<td>1-4</td>
</tr>
<tr>
<td>2–1</td>
<td>Manage IT Resource Page for Connector Server IT Resource</td>
<td>2-25</td>
</tr>
<tr>
<td>2–2</td>
<td>Edit IT Resource Details and Parameters Page for the Connector Server IT Resource</td>
<td>2-25</td>
</tr>
</tbody>
</table>
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 .................................................................................................. 2-12
2–3 Log Levels and ODL Message Type:Level Combinations .......................................................... 2-14
2–4 IT Resource Parameters ............................................................................................................ 2-21
2–5 Parameters of the IT Resource for the DBUM Connector Server ............................................. 2-26
2–6 Mappings for DBUM Oracle Event Handler/Adapter ................................................................. 2-32
3–1 Truststore Locations on Supported Application Servers .......................................................... 3-2
3–2 Lookup Definitions Synchronized with Oracle Database ............................................................ 3-5
3–3 Entries in Lookup.DBUM.oracle.AuthType Lookup Definition .................................................. 3-6
3–4 Entries in Lookup.DBUM.oracle.Configuration Lookup Definition ........................................... 3-6
3–5 Entries in Lookup.DBUM.MSSQL.Login.Configuration ............................................................... 3-7
3–6 Entries in Lookup.DBUM.MSSQL.UM.Configuration ................................................................. 3-7
3–7 Entries in the Lookup.DBUM.MSSQL.UM.Configuration.Trusted Lookup Definition .............. 3-8
3–8 Entries in Lookup.DBUM.MSSQL.UM.Configuration.Trusted .................................................. 3-8
3–9 Entries in Lookup.DBUM.MSSQL.Login.ProvAttrMap .............................................................. 3-8
3–10 Entries in Lookup.DBUM.MSSQL.Login.ReconAttrMap ........................................................... 3-9
3–11 Entries in Lookup.DBUM.MSSQL.UM.ProvAttrMap ................................................................. 3-10
3–12 Entries in Lookup.DBUM.MSSQL.UM.ReconAttrMap ............................................................... 3-10
3–13 Entries in Lookup.DBUM.MSSQL.UM.ReconDefaults ............................................................... 3-10
3–14 Entries in Lookup.DBUM.MSSQL.UM.ReconDefaults.Trusted .................................................. 3-10
3–15 Entries in Lookup.DBUM.MSSQL.UM.ReconAttrMap.Trusted .................................................. 3-11
3–16 Entries in Lookup.DBUM.MSSQL.Login.ReconTransformation .............................................. 3-12
3–17 Attributes of the Scheduled Jobs for Lookup Field Synchronization for MSSQL .................. 3-13
3–18 Attributes of the Scheduled Jobs for Reconciliation for MSSQL .............................................. 3-14
3–19 Attributes of the Scheduled Jobs for Delete Operations ......................................................... 3-15
3–20 Action Rules for Target Resource Reconciliation for MSSQL .................................................. 3-21
3–21 Action Rules for Trusted Source Reconciliation for MSSQL .................................................. 3-21
3–22 Script Section Logic for MSSQL Provisioning Queries ............................................................ 3-34
3–23 Connector Objects and Their Associations ........................................................................... 3-43
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 .............................................................................. 4-8
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.UM.Configuration.Trusted ..................................................... 4-10
4–7 Entries in Lookup.DBUM.oracle.UM.Configuration.Trusted ..................................................... 4-10
4–8 Entries in Lookup.DBUM.oracle.UM.ProvAttrMap ................................................................. 4-11
4–9 Entries in Lookup.DBUM.oracle.UM.ReconAttrMap ................................................................. 4-11
4–10 Entries in Lookup.DBUM.oracle.UM.ReconAttrMap.Trusted .................................................... 4-12
4–11 Entries in Lookup.DBUM.oracle.UM.ReconDefaults.Trusted .................................................... 4-12
4–12 Entries in Lookup.DBUM.oracle.UM.ReconTransformations ................................................... 4-13
4–13 Attributes of the Scheduled Jobs for Lookup Field Synchronization .................................... 4-15
4–14 Attributes of the Scheduled Jobs for Reconciliation ............................................................... 4-16
4–15 Attributes of the Scheduled Jobs for Delete Operations ......................................................... 4-16
4–16 Action Rules for Target Resource Reconciliation ................................................................. 4-23
4–17 Action Rules for Trusted Source Reconciliation ................................................................. 4-23
4–18 Script Section Logic for Oracle Provisioning Queries ............................................................ 4-37
4–19 Connector Objects and Their Associations ........................................................................... 4-46
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
<table>
<thead>
<tr>
<th>Page</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-5</td>
<td>Entries in Lookup.DBUM.MySQL.Configuration.Trusted</td>
<td>5-6</td>
</tr>
<tr>
<td>5-6</td>
<td>Entries in Lookup.DBUM.MySQL.UM.Configuration.Trusted</td>
<td>5-6</td>
</tr>
<tr>
<td>5-7</td>
<td>Entries in Lookup.DBUM.MySQL.UM.ProvAttrMap</td>
<td>5-7</td>
</tr>
<tr>
<td>5-8</td>
<td>Entries in Lookup.DBUM.MySQL.UM.ReconAttrMap</td>
<td>5-7</td>
</tr>
<tr>
<td>5-9</td>
<td>Entries in Lookup.DBUM.MySQL.UM.ReconAttrMap.Trusted</td>
<td>5-7</td>
</tr>
<tr>
<td>5-10</td>
<td>Entries in Lookup.DBUM.MySQL.UM.ReconDefaults.Trusted</td>
<td>5-8</td>
</tr>
<tr>
<td>5-11</td>
<td>Attributes of the Scheduled Job for Lookup Field Synchronization</td>
<td>5-10</td>
</tr>
<tr>
<td>5-12</td>
<td>Attributes of the Scheduled Jobs for Reconciliation</td>
<td>5-11</td>
</tr>
<tr>
<td>5-13</td>
<td>Attributes of the Scheduled Jobs for Delete Operations</td>
<td>5-11</td>
</tr>
<tr>
<td>5-14</td>
<td>Action Rules for Target Resource Reconciliation</td>
<td>5-17</td>
</tr>
<tr>
<td>5-15</td>
<td>Action Rules for Trusted Source Reconciliation</td>
<td>5-17</td>
</tr>
<tr>
<td>5-16</td>
<td>Script Section Logic for MySQL Provisioning Queries</td>
<td>5-29</td>
</tr>
<tr>
<td>5-17</td>
<td>Connector Objects and Their Associations</td>
<td>5-36</td>
</tr>
<tr>
<td>6-1</td>
<td>Truststore Locations on Supported Application Servers</td>
<td>6-2</td>
</tr>
<tr>
<td>6-2</td>
<td>Lookup Definitions Synchronized with DB2</td>
<td>6-4</td>
</tr>
<tr>
<td>6-3</td>
<td>Entries in Lookup.DBUM.DB2.Configuration</td>
<td>6-5</td>
</tr>
<tr>
<td>6-4</td>
<td>Entries in Lookup.DBUM.DB2.UM.Configuration</td>
<td>6-6</td>
</tr>
<tr>
<td>6-5</td>
<td>Entries in Lookup.DBUM.DB2.UM.Configuration.Trusted</td>
<td>6-6</td>
</tr>
<tr>
<td>6-6</td>
<td>Entries in Lookup.DBUM.DB2.UM.Configuration.Trusted</td>
<td>6-6</td>
</tr>
<tr>
<td>6-7</td>
<td>Entries in Lookup.DBUM.DB2.UM.ProvAttrMap</td>
<td>6-7</td>
</tr>
<tr>
<td>6-8</td>
<td>Entries in Lookup.DBUM.DB2.UM.ReconAttrMap</td>
<td>6-7</td>
</tr>
<tr>
<td>6-9</td>
<td>Entries in Lookup.DBUM.DB2.UM.ReconAttrMap.Trusted</td>
<td>6-8</td>
</tr>
<tr>
<td>6-10</td>
<td>Entries in Lookup.DBUM.DB2.UM.ReconDefaults.Trusted</td>
<td>6-8</td>
</tr>
<tr>
<td>6-11</td>
<td>Attributes of the Scheduled Jobs for Lookup Field Synchronization</td>
<td>6-10</td>
</tr>
<tr>
<td>6-12</td>
<td>Attributes of the Scheduled Jobs for Reconciliation</td>
<td>6-11</td>
</tr>
<tr>
<td>6-13</td>
<td>Action Rules for Target Resource Reconciliation</td>
<td>6-17</td>
</tr>
<tr>
<td>6-14</td>
<td>Action Rules for Trusted Source Reconciliation</td>
<td>6-17</td>
</tr>
<tr>
<td>6-15</td>
<td>Script Section Logic for DB2 Provisioning Queries</td>
<td>6-30</td>
</tr>
<tr>
<td>6-16</td>
<td>Connector Objects and Their Associations</td>
<td>6-37</td>
</tr>
<tr>
<td>7-1</td>
<td>Truststore Locations on Supported Application Servers</td>
<td>7-2</td>
</tr>
<tr>
<td>7-2</td>
<td>Lookup Definitions Synchronized with Sybase</td>
<td>7-4</td>
</tr>
<tr>
<td>7-3</td>
<td>Entries in Lookup.DBUM.Sybase.Configuration</td>
<td>7-5</td>
</tr>
<tr>
<td>7-4</td>
<td>Entries in Lookup.DBUM.Sybase.Login.Configuration</td>
<td>7-5</td>
</tr>
<tr>
<td>7-5</td>
<td>Entries in Lookup.DBUM.Sybase.UM.Configuration</td>
<td>7-6</td>
</tr>
<tr>
<td>7-6</td>
<td>Entries in the Lookup.DBUM.Sybase.UM.Configuration.Trusted</td>
<td>7-6</td>
</tr>
<tr>
<td>7-7</td>
<td>Entries in Lookup.DBUM.Sybase.UM.Configuration.Trusted</td>
<td>7-6</td>
</tr>
<tr>
<td>7-8</td>
<td>Entries in Lookup.DBUM.Sybase.Login.ProvAttrMap</td>
<td>7-7</td>
</tr>
<tr>
<td>7-9</td>
<td>Entries in Lookup.DBUM.Sybase.Login.ReconAttrMap</td>
<td>7-7</td>
</tr>
<tr>
<td>7-10</td>
<td>Entries in Lookup.DBUM.Sybase.Login.ReconDefaults</td>
<td>7-8</td>
</tr>
<tr>
<td>7-11</td>
<td>Entries in Lookup.DBUM.Sybase.UM.ProvAttrMap</td>
<td>7-8</td>
</tr>
<tr>
<td>7-12</td>
<td>Entries in Lookup.DBUM.Sybase.UM.ReconAttrMap</td>
<td>7-8</td>
</tr>
<tr>
<td>7-13</td>
<td>Entries in Lookup.DBUM.Sybase.UM.ReconDefaults</td>
<td>7-9</td>
</tr>
<tr>
<td>7-14</td>
<td>Entries in Lookup.DBUM.Sybase.UM.ReconDefaults.Trusted</td>
<td>7-9</td>
</tr>
<tr>
<td>7-15</td>
<td>Entries in Lookup.DBUM.Sybase.UM.ReconAttrMap.Trusted</td>
<td>7-9</td>
</tr>
<tr>
<td>7-16</td>
<td>Entries in Lookup.DBUM.Sybase.Login.ReconTransformations</td>
<td>7-10</td>
</tr>
<tr>
<td>7-17</td>
<td>Attributes of the Scheduled Jobs for Lookup Field Synchronization</td>
<td>7-11</td>
</tr>
<tr>
<td>7-18</td>
<td>Attributes of the Scheduled Jobs for Reconciliation</td>
<td>7-13</td>
</tr>
<tr>
<td>7-19</td>
<td>Attributes of the Scheduled Jobs for Delete Operations</td>
<td>7-13</td>
</tr>
<tr>
<td>7-20</td>
<td>Action Rules for Target Resource Reconciliation</td>
<td>7-16</td>
</tr>
<tr>
<td>7-21</td>
<td>Script Section Logic for Sybase Provisioning Queries</td>
<td>7-28</td>
</tr>
<tr>
<td>7-22</td>
<td>Connector Objects and Their Associations</td>
<td>7-36</td>
</tr>
<tr>
<td>8-1</td>
<td>IT Resource Parameters for New Database</td>
<td>8-6</td>
</tr>
</tbody>
</table>
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**


**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](http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info) or visit [http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs](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, see the Oracle Identity Manager documentation library.

For generic information about connectors, see Oracle Fusion Middleware User’s Guide for Oracle Identity Manager.

The following Oracle Technology Network page provides links to Oracle Identity Manager documentation:

[http://docs.oracle.com/cd/E14571_01/im.htm](http://docs.oracle.com/cd/E14571_01/im.htm)

**Documentation Updates**

Oracle is committed to delivering the best and most recent information available. For information about updates to the Oracle Identity Manager Connectors documentation library, visit Oracle Technology Network at [http://docs.oracle.com/cd/E22999_01/index.htm](http://docs.oracle.com/cd/E22999_01/index.htm)
## Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td><strong>monospace</strong></td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
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:
<table>
<thead>
<tr>
<th>Bug Number</th>
<th>Issue</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>16476930</td>
<td>The Entitlement, AccountName, and AccountID properties of fields on the process form for all target systems supported by this connector were not tagged.</td>
<td>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 true.</td>
</tr>
<tr>
<td>16476913</td>
<td>The &quot;Entitlement&quot; property of child form attributes was not set to true. 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. The &quot;AccountName&quot; property of a process form field that represented the login ID of an account in the target system was not set to true. 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. The &quot;AccountID&quot; property of a process form field that represented the immutable GUID of the account (if one exists) was not set to true. This prevented integrating Oracle Identity Manager with Oracle Identity Analytics.</td>
<td></td>
</tr>
<tr>
<td>16491431</td>
<td>The following issue was observed if you are using MySQL as the target system: 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 &quot;Waiting&quot; state.</td>
<td>This issue has been resolved.</td>
</tr>
<tr>
<td>15955807</td>
<td>The connector failed to update password in UpdateApiOp with MySQL database. In addition, no errors were reported.</td>
<td>This issue has been resolved. The connector updates the password in UpdateApiOp with MySQL database.</td>
</tr>
<tr>
<td>16085209</td>
<td>The following exception was encountered when you ran the connector test: DBUM Connector Test Fails JRF Portability Layer exception</td>
<td>This issue has been resolved.</td>
</tr>
</tbody>
</table>
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 Chapter 5, "Using and Extending the Connector for MySQL" for more information.
- DB2
  See Chapter 6, "Using and Extending the Connector for DB2" for more information.
- Sybase
  See Chapter 7, "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

<table>
<thead>
<tr>
<th>Bug Number</th>
<th>Issue</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>15919151</td>
<td>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.</td>
<td>This issue has been resolved.</td>
</tr>
<tr>
<td>15895008</td>
<td>Changing the password of a user account in Sybase resulted in the failure of a stored procedure.</td>
<td>This issue has been resolved.</td>
</tr>
</tbody>
</table>

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.
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 the "Understanding the Identity Connector Framework" chapter in the Oracle Fusion Middleware Developer’s Guide 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 Section 1.4, "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 Section 1.5.9, "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 Section 2.2.2.3, "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:

- Section 2.3.3, "Configuring the Connector to Support Multiple Versions of the Target System"
- Section 4.7.3, "Configuring the Connector for Multiple Installations of Oracle Database"
- Section 3.7.3, "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:
■ Section 4.6, "Provisioning for Oracle Database"
■ Section 3.6, "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:
■ Section 4.7.6, "Configuring Transformation of Data During User Reconciliation for
Oracle Database"
■ Section 4.7.5, "Configuring Validation of Data During Reconciliation and
Provisioning for Oracle Database"
■ Section 3.7.6, "Configuring Transformation of Data During User Reconciliation for
MSSQL"
■ Section 3.7.5, "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:
■ Section 4.7.7, "Configuring Resource Exclusion Lists for Oracle Database"
■ Section 3.7.7, "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 Section 3.5, "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 Chapter 8, "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 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, "Certified Components".
- The "JDK" row of Table 1–1, "Certified Components" has been renamed to "Connector Server JDK".
- Section 2.5.1, "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, "Certified Components" has been updated.
- The "Target systems" row of Table 1–1, "Certified Components" has been updated.
- Information specific to Oracle Identity Manager 11g Release 2 PS3 (11.1.2.3.0) has been added to Section 1.2, "Usage Recommendation."
- The "Groovy" sample value has been removed from Step 3 of the following sections:
  - Section 3.7.8, "Configuring Action Scripts for MSSQL"
  - Section 4.7.8, "Configuring Action Scripts for Oracle Database"
  - Section 5.7.8, "Configuring Action Scripts for MySQL"
  - Section 6.7.8, "Configuring Action Scripts for DB2"
  - Section 7.7.8, "Configuring 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 Section 8.1, "Target System Attributes and Queries."
  - Information specific to "UPDATE_DBID" query has been added to the information in the "Provisioning.queries" file in Section 8.2, "Configuring the Queries."
  - The sample screenshot of the updated process task has been modified in Section 8.7, "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 Section 8.8, "Adding Attributes for Reconciliation."
- A "Note" with information specific to lookup queries has been added at the beginning of the following sections:
  - Section 3.7, "Extending the Connector for MSSQL"
  - Section 4.7, "Extending the Connector for Oracle Database"
Section 5.7, "Extending the Connector for MySQL"
Section 6.7, "Extending the Connector for DB2"
Section 7.7, "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, "Certified Components".
■ Information about Oracle 12c has been added to the "JDBC URL" row of Table 2–4, "IT Resource Parameters".
■ Information about Microsoft SQL Server 2012 has been added to Section 2.3.3, "Configuring the Connector to Support Multiple Versions of the Target System."
■ Information about limited reconciliation has been modified in the following sections:
  – Section 3.5.9, "Performing Limited Reconciliation from MSSQL"
  – Section 4.5.9, "Performing Limited Reconciliation from Oracle Database"
  – Section 5.5.8, "Performing Limited Reconciliation from MySQL"
  – Section 6.5.8, "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 Section 2.4.3, "Postupgrade Steps".
■ Information was updated under bug number 16713445 in Chapter 10, "Known Issues and Workarounds".
■ Information about exclusion lists has been modified in the following sections:
  – Section 4.7.7, "Configuring Resource Exclusion Lists for Oracle Database"
  – Section 3.7.7, "Configuring Resource Exclusion Lists for MSSQL"
  – Section 5.7.7, "Configuring Resource Exclusion Lists for MySQL"
  – Section 6.7.7, "Configuring Resource Exclusion Lists for DB2"
  – Section 7.7.7, "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:
  – Section 9.1, "Running the Test Utility for the Oracle Database"
  – Section 9.2, "Running the Test Utility for the MSSQL Database"
  – Section 9.3, "Running the Test Utility for MySQL"
  – Section 9.4, "Running the Test Utility for DB2"
  – Section 9.5, "Running the Test Utility for Sybase"
■ The name of the "Known Issues" chapter has been changed to "Known Issues and Workarounds." In addition, Chapter 10, "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 Section 1–1, "Certified Components" has been modified.
A note has been added in the "Files in the xml directory" row of Table 2–1, "Files and Directories on the Installation Media".

The following sections have been added:

- Section 1.2, "Usage Recommendation"
- Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later"
- Section 2.3.1.8, "Localizing Field Labels in UI Forms"
- Section 4.6.6, "Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x"
- Section 3.6.6, "Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x"
- Section 5.6.6, "Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x"
- Section 6.6.6, "Performing Provisioning Operations in Oracle Identity Manager Release 11.1.2.x"
- Section 7.6.6, "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:

- Section 2.2.1, "Installing the Connector in Oracle Identity Manager"
- Section 2.3.2, "Configuring the IT Resource for the Target System"
- Section 2.3.4, "Configuring the IT Resource for the Connector Server"
- Section 4.4.3, "Configuring Scheduled Jobs for Oracle Database"
- Section 4.5.5, "Configuring the Target System As a Trusted Source"
- Section 4.7.2.2, "Configuring Oracle Identity Manager"
- Section 3.4.3, "Configuring Scheduled Jobs for MSSQL"
- Section 3.5.5, "Configuring the Target System As a Trusted Source"
- Section 3.7.2.2, "Configuring Oracle Identity Manager"
- Section 5.4.3, "Configuring Scheduled Jobs for MySQL"
- Section 5.5.4, "Configuring the Target System As a Trusted Source"
- Section 5.7.2.2, "Configuring Oracle Identity Manager"
- Section 6.4.3, "Configuring Scheduled Jobs for DB2"
- Section 6.5.4, "Configuring the Target System As a Trusted Source"
- Section 6.7.2.2, "Configuring Oracle Identity Manager"
- Section 7.4.3, "Configuring Scheduled Jobs for Sybase"
- Section 7.7.2.2, "Configuring Oracle Identity Manager"
- Section 8.4, "Configuring the IT Resource"

Issues tracked by bug numbers 16713445 and 16482139 have been added to Chapter 10, "Known Issues and Workarounds."

Bug 14260504 has been removed from Chapter 10, "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, "Certified Components":
  - 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.
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.

---

This chapter contains the following sections:

- Section 1.1, "Certified Components"
- Section 1.2, "Usage Recommendation"
- Section 1.3, "Certified Languages"
- Section 1.4, "Connector Architecture"
- Section 1.5, "Features of the Connector"
- Section 1.6, "Roadmap for Deploying and Using the Connector"
1.1 Certified Components

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

<table>
<thead>
<tr>
<th>Component</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Identity Manager</td>
<td>You can use one of the following releases of Oracle Identity Manager:</td>
</tr>
<tr>
<td></td>
<td>- Oracle Identity Manager release 11g PS1 (11.1.1.5.3) BP03 and any later BP in</td>
</tr>
<tr>
<td></td>
<td>this release track</td>
</tr>
<tr>
<td></td>
<td>If you are using Oracle Identity Manager 11.1.1.5.3, then you must download</td>
</tr>
<tr>
<td></td>
<td>and apply the patches 14163597 and 13897038. To download a patch, sign in to</td>
</tr>
<tr>
<td></td>
<td>My Oracle Support and search for the patch number on the Patches and</td>
</tr>
<tr>
<td></td>
<td>Updates page at:</td>
</tr>
<tr>
<td></td>
<td><a href="https://support.oracle.com/">https://support.oracle.com/</a></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> In this guide, Oracle Identity Manager release 11.1.1 has been used</td>
</tr>
<tr>
<td></td>
<td>to denote Oracle Identity Manager release 11.1.1.5 BP03, and future releases</td>
</tr>
<tr>
<td></td>
<td>in the 11.1.1 series that the connector will support.</td>
</tr>
<tr>
<td></td>
<td>- Oracle Identity Manager 11g Release 2 (11.1.2.0.4) and any later BP in this</td>
</tr>
<tr>
<td></td>
<td>release track</td>
</tr>
<tr>
<td></td>
<td>- Oracle Identity Manager 11g Release 2 PS3 (11.1.2.3.0)</td>
</tr>
<tr>
<td>Target systems</td>
<td>The target system can be any one of the following:</td>
</tr>
<tr>
<td></td>
<td>- Exadata V2</td>
</tr>
<tr>
<td></td>
<td>- Oracle9i Database</td>
</tr>
<tr>
<td></td>
<td>- Oracle Database 10g and 11g as either single database or Oracle RAC</td>
</tr>
<tr>
<td></td>
<td>implementation</td>
</tr>
<tr>
<td></td>
<td>- Oracle Database 12c as single database, pluggable database (PDB), or Oracle</td>
</tr>
<tr>
<td></td>
<td>RAC implementation</td>
</tr>
<tr>
<td></td>
<td>- MySQL 5.x</td>
</tr>
<tr>
<td></td>
<td>- IBM DB2 UDB 9.x</td>
</tr>
<tr>
<td></td>
<td>- Sybase 15.x</td>
</tr>
<tr>
<td>Connector Server</td>
<td>11.1.2.1.0</td>
</tr>
<tr>
<td>Connector Server JDK</td>
<td>JDK 1.6 or later, or JRockit JDK 1.6 or later.</td>
</tr>
<tr>
<td>Target system user account</td>
<td>Depending on the target system, one of the following user accounts is used by</td>
</tr>
<tr>
<td></td>
<td>Oracle Identity Manager to perform reconciliation and provisioning operations on</td>
</tr>
<tr>
<td></td>
<td>the target system:</td>
</tr>
<tr>
<td></td>
<td>- For Oracle Database: sys as sysdba, or system</td>
</tr>
<tr>
<td></td>
<td>- For Microsoft SQL Server: sa (administrator)</td>
</tr>
<tr>
<td></td>
<td>- For MySQL: root</td>
</tr>
<tr>
<td></td>
<td>- For DB2: db2admin</td>
</tr>
<tr>
<td></td>
<td>- For Sybase: sa (administrator)</td>
</tr>
</tbody>
</table>

1.2 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), then you must use the latest 11.1.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.

1.3 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
1.4 Connector Architecture

This 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:** The "Understanding the Identity Connector Framework" chapter in Oracle Fusion Middleware Developer’s Guide 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. See the following sections for more details about lookup definitions for each database:

- Section 4.3, "Lookup Definitions for Oracle Database"
- Section 3.3, "Lookup Definitions for MSSQL"

**Note:** However, the connector does not support the entry of multibyte characters in some of the fields.
Section 5.3, "Lookup Definitions for MySQL"

Section 6.3, "Lookup Definitions for DB2"

Section 7.3, "Lookup Definitions for Sybase"

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

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

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:

```java
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, while 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:

```java
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:

```java
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.

1.5 Features of the Connector

The following are features of the connector:

- Section 1.5.1, "Mapping Standard and Custom Attributes for Reconciliation and Provisioning"
- Section 1.5.2, "Predefined and Custom Reconciliation Queries"
- Section 1.5.3, "Predefined and Custom Provisioning Queries"
- Section 1.5.4, "Support for Creating Global and External Users In Oracle Database"
- Section 1.5.5, "Full and Incremental Reconciliation"
- Section 1.5.6, "Limited (Filtered) Reconciliation"
- Section 1.5.7, "Batched Reconciliation"
- Section 1.5.8, "Specifying Accounts to Be Excluded from Reconciliation and Provisioning Operations"
- Section 1.5.9, "Connection Pooling"
- Section 1.5.10, "Support for Connector Server"
- Section 1.5.11, "Support for Creating Connector Copies"
- Section 1.5.12, "Transformation and Validation of Account Data"
- Section 1.5.13, "Support for Reconciling Data About Deleted Entities"
- Section 1.5.14, "Separate Scheduled Jobs for Reconciliation of Users, Logins, and Deleted Login Entities"
- Section 1.5.15, "Support for SSL Communication Between the Target System and Oracle Identity Manager"
- Section 1.5.16, "Support for Managing Authorization to Oracle Database Vault Realms"
- Section 1.5.17, "Support for Configuring the Connector for Enterprise User Security"
1.5.1 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:
- Section 4.7, "Extending the Connector for Oracle Database"
- Section 3.7, "Extending the Connector for MSSQL"
- Section 5.7, "Extending the Connector for MySQL"
- Section 6.7, "Extending the Connector for DB2"
- Section 7.7, "Extending the Connector for Sybase"

1.5.2 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:
- Section 4.7.1.2, "Syntax of Reconciliation Queries for Oracle Database"
- Section 3.7.1.2, "Syntax of Reconciliation Queries for MSSQL"
- Section 5.7.1.2, "Syntax of Reconciliation Queries for MySQL Database"
- Section 6.7.1.2, "Syntax of Reconciliation Queries for DB2 Database"
- Section 7.7.1.2, "Syntax of Reconciliation Queries for Sybase Database"

1.5.3 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:
- Section 4.7.1.1, "Syntax of Provisioning Queries for Oracle Database"
- Section 3.7.1.1, "Syntax of Provisioning Queries for MSSQL"
- Section 5.7.1.1, "Syntax of Provisioning Queries for MySQL Database"
- Section 6.7.1.1, "Syntax of Provisioning Queries for DB2 Database"
1.5.4 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:

- Section 1.5.17, "Support for Configuring the Connector for Enterprise User Security"
- Section 4.6.1, "Guidelines on Performing Provisioning Operations for Oracle Database"

1.5.5 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:

- Section 4.5, "Reconciliation from Oracle Database"
- Section 3.5, "Reconciliation from MSSQL"
- Section 5.5, "Reconciliation from MySQL"
- Section 6.5, "Reconciliation from DB2"
- Section 7.5, "Reconciliation from Sybase"

1.5.6 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:

- Section 4.5.9, "Performing Limited Reconciliation from Oracle Database"
- Section 3.5.9, "Performing Limited Reconciliation from MSSQL"
- Section 5.5.8, "Performing Limited Reconciliation from MySQL"
- Section 6.5.8, "Performing Limited Reconciliation from DB2"
1.5.7 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:

- Section 4.5.10, "Performing Batched Reconciliation from Oracle Database"
- Section 5.5.9, "Performing Batched Reconciliation from MySQL"
- Section 6.5.9, "Performing Batched Reconciliation from DB2"

1.5.8 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:

- Section 4.7.7, "Configuring Resource Exclusion Lists for Oracle Database"
- Section 3.7.7, "Configuring Resource Exclusion Lists for MSSQL"
- Section 5.7.7, "Configuring Resource Exclusion Lists for MySQL"
- Section 6.7.7, "Configuring Resource Exclusion Lists for DB2"
- Section 7.7.7, "Configuring Resource Exclusion Lists for Sybase"

1.5.9 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. **Section 2.3.1.5, "Setting up the Lookup Definition for Connection Pooling"** provides information about setting up the connection pool.
1.5.10 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 Section 2.2.2.1, "Installing and Configuring the Connector Server" for more information.

1.5.11 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:

- Section 4.7.3, "Configuring the Connector for Multiple Installations of Oracle Database"
- Section 3.7.3, "Configuring the Connector for Multiple Installations of MSSQL"
- Section 5.7.3, "Configuring the Connector for Multiple Installations of MySQL"
- Section 6.7.3, "Configuring the Connector for Multiple Installations of DB2"
- Section 7.7.3, "Configuring the Connector for Multiple Installations of Sybase"

1.5.12 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:

- Section 4.7.5, "Configuring Validation of Data During Reconciliation and Provisioning for Oracle Database"
- Section 4.7.6, "Configuring Transformation of Data During User Reconciliation for Oracle Database"

For MSSQL:

- Section 3.7.5, "Configuring Validation of Data During Reconciliation and Provisioning for MSSQL"
- Section 3.7.6, "Configuring Transformation of Data During User Reconciliation for MSSQL"

For MySQL:

- Section 5.7.5, "Configuring Validation of Data During Reconciliation and Provisioning for MySQL"
- Section 5.7.6, "Configuring Transformation of Data During User Reconciliation for MySQL"

For DB2:
### 1.5.13 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.

### 1.5.14 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:

- Section 4.4.3, "Configuring Scheduled Jobs for Oracle Database"
- Section 3.4.3, "Configuring Scheduled Jobs for MSSQL"
- Section 5.4.3, "Configuring Scheduled Jobs for MySQL"
- Section 6.4.3, "Configuring Scheduled Jobs for DB2"
- Section 7.4.3, "Configuring Scheduled Jobs for Sybase"

### 1.5.15 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:

- Section 4.1, "Configuring Secure Communication Between Oracle Database and Oracle Identity Manager"
- Section 3.1, "Configuring Secure Communication Between MSSQL and Oracle Identity Manager"
- Section 5.1, "Configuring Secure Communication Between MySQL and Oracle Identity Manager"
- Section 6.1, "Configuring Secure Communication Between DB2 and Oracle Identity Manager"
1.5.16 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 (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 Section 2.3.1.4, "Creating the Administrator Account on Oracle Database Vault" for more information.

1.5.17 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.

### 1.6 Roadmap for Deploying and Using the Connector

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

- **Chapter 2, "Deploying the Connector"** describes procedures that you must perform on Oracle Identity Manager and the target system during each stage of connector deployment.

- **Chapter 4, "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.

- **Chapter 3, "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.

- **Chapter 5, "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.

- **Chapter 6, "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.

- **Chapter 7, "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.

- **Section 8, "Configuring the Connector for a JDBC-Based Database"** describes procedures for creating the connector for the JDBC-based database.

- **Chapter 9, "Testing the Connector"** describes procedures to test the connector.

- **Chapter 10, "Known Issues and Workarounds"** lists known issues associated with this release of the connector.
Deploying the Connector

The procedure to deploy the connector can be divided into the following stages:

- Section 2.1, "Preinstallation"
- Section 2.2, "Installation"
- Section 2.3, "Postinstallation"
- Section 2.4, "Upgrading the Connector"
- Section 2.5, "Postcloning Steps"

2.1 Preinstallation

Preinstallation information is divided across the following sections:

- Section 2.1.1, "Files and Directories on the Installation Media"
- Section 2.1.2, "Preinstallation on Microsoft SQL Server"

2.1.1 Files and Directories on the Installation Media

Table 2–1 describes the files and directories on the installation media.

<table>
<thead>
<tr>
<th>File in the Installation Media Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File in the bundle directory:</td>
<td></td>
</tr>
<tr>
<td>org.identityconnectors.dbum-1.0.1116.jar</td>
<td></td>
</tr>
<tr>
<td>This file contains connector code, SQL queries, and stored procedures that are used for provisioning and reconciliation.</td>
<td></td>
</tr>
<tr>
<td>Files in the configuration directory:</td>
<td></td>
</tr>
<tr>
<td>DBUM-Oracle-CI.xml</td>
<td></td>
</tr>
<tr>
<td>DBUM-MSSQL-CI.xml</td>
<td></td>
</tr>
<tr>
<td>DBUM-MySQL-CI.xml</td>
<td></td>
</tr>
<tr>
<td>DBUM-DB2-CI.xml</td>
<td></td>
</tr>
<tr>
<td>DBUM-Sybase-CI.xml</td>
<td></td>
</tr>
<tr>
<td>This directory contains the configuration files that are used by the Connector Installer during installation of the connector for a particular target system.</td>
<td></td>
</tr>
<tr>
<td>Files in the javadoc directory</td>
<td></td>
</tr>
<tr>
<td>This directory contains information about the Java APIs used by the connector.</td>
<td></td>
</tr>
<tr>
<td>File in the lib directory:</td>
<td></td>
</tr>
<tr>
<td>DBUM-oim-integration.jar</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>
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 are displayed on the Administrative and User Console. These text strings 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

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.

- 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

**Note:** The dataset XML files are applicable only if you are using Oracle Identity Manager release 11.1.1.x.

---

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

<table>
<thead>
<tr>
<th>File in the Installation Media Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Files in the resources directory</td>
<td>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. <strong>Note:</strong> A <strong>resource bundle</strong> is a file containing localized versions of the text strings that are displayed on the Administrative and User Console. These text strings include GUI element labels and messages.</td>
</tr>
<tr>
<td>Files in the test directory:</td>
<td>This directory contains the files for testing the connector.</td>
</tr>
<tr>
<td>config\oracleconfig.properties</td>
<td></td>
</tr>
<tr>
<td>config\mssqlconfig.properties</td>
<td></td>
</tr>
<tr>
<td>config\mysqlconfig.properties</td>
<td></td>
</tr>
<tr>
<td>config\db2config.properties</td>
<td></td>
</tr>
<tr>
<td>config\sybaseconfig.properties</td>
<td></td>
</tr>
<tr>
<td>lib\DBUMTest.jar</td>
<td></td>
</tr>
<tr>
<td>scripts\DBUMProvisioningTester.bat</td>
<td></td>
</tr>
<tr>
<td>scripts\DBUMProvisioningTester.sh</td>
<td></td>
</tr>
<tr>
<td>thirdparty (folder)</td>
<td></td>
</tr>
<tr>
<td>Files in the upgrade directory:</td>
<td>This directory contains the scripts for performing the post-upgrade operations.</td>
</tr>
<tr>
<td>PostUpgradeScriptOracleDBUM.sql</td>
<td></td>
</tr>
<tr>
<td>PostUpgradeScriptMSSQLDBUM.sql</td>
<td></td>
</tr>
<tr>
<td>PostUpgradeScriptMySQLDBUM.sql</td>
<td></td>
</tr>
<tr>
<td>PostUpgradeScriptDB2DBUM.sql</td>
<td></td>
</tr>
<tr>
<td>PostUpgradeScriptSybaseDBUM.sql</td>
<td></td>
</tr>
<tr>
<td>Files in the xml directory:</td>
<td>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.</td>
</tr>
<tr>
<td>DBUserManagement-Oracle-ConnectorConfig.xml</td>
<td></td>
</tr>
<tr>
<td>DBUserManagement-Oracle-Datasets.xml</td>
<td></td>
</tr>
<tr>
<td>DBUserManagement-MSSQL-ConnectorConfig.xml</td>
<td></td>
</tr>
<tr>
<td>DBUserManagement-MSSQL-Datasets.xml</td>
<td></td>
</tr>
<tr>
<td>DBUserManagement-MySQL-ConnectorConfig.xml</td>
<td></td>
</tr>
<tr>
<td>DBUserManagement-MySQL-Datasets.xml</td>
<td></td>
</tr>
<tr>
<td>DBUserManagement-DB2-ConnectorConfig.xml</td>
<td></td>
</tr>
<tr>
<td>DBUserManagement-DB2-Datasets.xml</td>
<td></td>
</tr>
<tr>
<td>DBUserManagement-Sybase-ConnectorConfig.xml</td>
<td></td>
</tr>
<tr>
<td>DBUserManagement-Sybase-Datasets.xml</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The dataset XML files are applicable only if you are using Oracle Identity Manager release 11.1.1.x.
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.

2.2 Installation

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 Section 2.2.1, "Installing the Connector in Oracle Identity Manager."

- To run the connector code remotely in a Connector Server, perform the procedures described in Section 2.2.1, "Installing the Connector in Oracle Identity Manager" and Section 2.2.2, "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.
2.2.1 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. Depending on the Oracle Identity Manager release you are using, perform one of the following steps:

   - For Oracle Identity Manager release 11.1.1.x:
     a. Log in to the Administrative and User Console by using the user account described in the "Creating the User Account for Installing Connectors" section Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager.
     b. On the Welcome to Identity Manager Advanced Administration page, in the System Management region, click **Manage Connector**.

   - For Oracle Identity Manager release 11.2.x or later:
     a. Log in to Oracle Identity System Administration by using the user account described in the "Creating the User Account for Installing Connectors" section Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager.
     b. In the left pane, under System Management, click **Manage Connector**.

4. In the Manage Connector page, click **Install**.

5. 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

6. Click Load.

7. 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.

8. 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 Section 2.3.1.3, “Clearing Content Related to 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 Section 2.3.2, "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 Section 4.4.3, "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.

### 2.2.2 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 Section 2.2.1, "Installing the Connector in Oracle Identity Manager."
- See Section 2.3.4, "Configuring the IT Resource for the Connector Server" for related information.

This procedure can be divided into the following stages:

- Section 2.2.2.1, "Installing and Configuring the Connector Server"
- Section 2.2.2.2, "Running the Connector Server"
- Section 2.2.2.3, "Installing the Connector on the Connector Server"

#### 2.2.2.1 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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorserver.port</td>
<td>Port on which the Java Connector Server listens for requests. Default is 8763.</td>
</tr>
<tr>
<td>connectorserver.bundleDir</td>
<td>Directory where the connector bundles are deployed. Default is bundles.</td>
</tr>
<tr>
<td>connectorserver.libDir</td>
<td>Directory in which to place dependent libraries. Default is lib.</td>
</tr>
<tr>
<td>connectorserver.usessl</td>
<td>If set to true, the Java Connector Server uses SSL for secure communication. Default is false.</td>
</tr>
</tbody>
</table>

   If you specify true, use the following options on the command line when you start the Java Connector Server:

   -Djavax.net.ssl.keyStore
   -Djavax.net.ssl.keyStoreType (optional)
   -Djavax.net.ssl.keyStorePassword

<table>
<thead>
<tr>
<th>connectorserver.ifaddress</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectorserver.key</td>
<td>Java Connector Server key.</td>
</tr>
</tbody>
</table>

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 Section 2.2.2.2, "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.
2.2.2.2 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 Section 2.2.2.1, "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:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/install [serviceName] [&quot;-J javaoption&quot;]</td>
<td>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 ConnectorServerJava.</td>
</tr>
<tr>
<td>/run [&quot;-J javaoption&quot;]</td>
<td>Runs the Java Connector Server from the console. Optionally, you can specify Java options. For example, to run the Java Connector Server with SSL: ConnectorServer.bat /run &quot;-J-Djavax.net.ssl.keyStore=mykeystore.jks&quot; &quot;-J-Djavax.net.ssl.keyStorePassword=javax.net.ssl.keyStorePassword= password&quot;</td>
</tr>
<tr>
<td>/setKey [key]</td>
<td>Sets the Java Connector Server key. The ConnectorServer.bat script stores the hashed value of the key in the connectorserver.key property in the ConnectorServer.properties file.</td>
</tr>
<tr>
<td>/uninstall [serviceName]</td>
<td>Uninstalls the Java Connector Server. If you do not specify a service name, the script uninstalls the ConnectorServerJava service.</td>
</tr>
</tbody>
</table>

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

2.2.2.3 Installing the Connector on the Connector Server

See Also: "Using an Identity Connector Server" section in the Oracle Fusion Middleware Developer’s Guide 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:

- Section 2.3.1, "Postinstallation on Oracle Identity Manager"
- Section 2.3.2, "Configuring the IT Resource for the Target System"
- Section 2.3.3, "Configuring the Connector to Support Multiple Versions of the Target System"
- Section 2.3.4, "Configuring the IT Resource for the Connector Server"
- Section 2.3.5, "Enabling Logging for Connector Server"

### 2.3.1 Postinstallation on Oracle Identity Manager

This section discusses the following topics:

- Section 2.3.1.1, "Configuring the Target System As a Trusted Source"
- Section 2.3.1.2, "Changing to the Required Input Locale"
- Section 2.3.1.3, "Clearing Content Related to Connector Resource Bundles from the Server Cache"
- Section 2.3.1.4, "Creating the Administrator Account on Oracle Database Vault"
- Section 2.3.1.5, "Setting up the Lookup Definition for Connection Pooling"
- Section 2.3.1.6, "Enabling Logging for Oracle Identity Manager"
- Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later"
- Section 2.3.1.8, "Localizing Field Labels in UI Forms"

### 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.
The target system can be designated as a trusted source or target resource. As discussed earlier in this guide, 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.

If you designate the target system as a target resource, then during a reconciliation run:

- For each account created on the target system, a resource is assigned to the corresponding OIM User.
- Updates made to each account on the target system are propagated to the corresponding resource.

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

See Also: Section 2.3.2, "Configuring the IT Resource for the Target System" for more information about configuring the IT resource for the target system

For Oracle:

1. Create a new IT resource, for example, Oracle DB Trusted of type Oracle DBUM.
2. In the Configuration Lookup, update the trusted configuration lookup name as Lookup.DBUM.Oracle.Configuration.Trusted.

For MSSQL:

1. Create a new IT resource, for example, MSSQL DB Trusted of type MSSQL DBUM.
2. In the Configuration Lookup, update the trusted configuration lookup name as Lookup.DBUM.MSSQL.Configuration.Trusted.

For MySQL:

1. Create a new IT resource, for example, MySQL DB Trusted of type MySQL DBUM.
2. In the Configuration Lookup, update the trusted configuration lookup name as Lookup.DBUM.MySQL.Configuration.Trusted.

For DB2:

1. Create a new IT resource, for example, DB2 DB Trusted of type DB2 DBUM.
2. In the Configuration Lookup, update the trusted configuration lookup name as Lookup.DBUM.DB2.Configuration.Trusted.

For Sybase:

1. Create a new IT resource, for example, Sybase DB Trusted of type Sybase DBUM.
2. In the Configuration Lookup, update the trusted configuration lookup name as 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 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, perform one of the following steps:
   - 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.
See Oracle Fusion Middleware System Administrator’s Guide for Oracle Identity Manager for more information about the PurgeCache utility.

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:
   
   ```sql
   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:
   
   ```sql
   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):

   ```sql
   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>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool Max Idle</td>
<td>Maximum number of idle objects in a pool.</td>
</tr>
<tr>
<td></td>
<td>Default value: 10</td>
</tr>
</tbody>
</table>
Deploying the Connector

2.1.3 Modifying the Connection Pooling Properties

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.


   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 Enabling Logging for Oracle Identity Manager

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. Oracle Identity Manager uses Oracle Java Diagnostic Logging (OJDL) for logging. OJDL 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.

### Table 2–2 (Cont.) Connection Pooling Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pool Max Size</td>
<td>Maximum number of connections that the pool can create. Default value: 10</td>
</tr>
<tr>
<td>Pool Max Wait</td>
<td>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</td>
</tr>
<tr>
<td>Pool Min Evict Idle Time</td>
<td>Minimum time, in milliseconds, the connector must wait before evicting an idle object. Default value: 120000</td>
</tr>
<tr>
<td>Pool Min Idle</td>
<td>Minimum number of idle objects in a pool. Default value: 1</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>JAVA Level</th>
<th>ODL Message Type:Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEVERE.intValue()+100</td>
<td>INCIDENT_ERROR:1</td>
</tr>
<tr>
<td>SEVERE</td>
<td>ERROR:1</td>
</tr>
<tr>
<td>WARNING</td>
<td>WARNING:1</td>
</tr>
<tr>
<td>INFO</td>
<td>NOTIFICATION:1</td>
</tr>
<tr>
<td>CONFIG</td>
<td>NOTIFICATION:16</td>
</tr>
<tr>
<td>FINE</td>
<td>TRACE:1</td>
</tr>
<tr>
<td>FINER</td>
<td>TRACE:16</td>
</tr>
<tr>
<td>FINEST</td>
<td>TRACE:32</td>
</tr>
</tbody>
</table>

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.

To enable logging in Oracle WebLogic Server:

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

```xml
<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"/>
```
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]:

```xml
<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

If you are using Oracle Identity Manager release 11.1.2 or later, you must create additional metadata such as UI forms and application instances. In addition, you must run the entitlement and catalog sync jobs.

1. Log in to Oracle Identity System Administration.

2. Create and activate a sandbox as follows. For detailed instructions, see the "Managing Sandboxes" section in the Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager.

   a. In the upper right corner of the page, click the Sandboxes link.

   The Manage Sandboxes page is displayed.

   b. On the toolbar, click Create Sandbox.
c. In the Create Sandbox dialog box, enter values for the following fields:
   - **Sandbox Name**: Enter a name for the sandbox.
   - **Sandbox Description**: Enter a description of the sandbox.

d. Click **Save and Close**.

e. Click **OK** on the confirmation message that is displayed.
   The sandbox is created and displayed in the Available Sandboxes section of the Manage Sandboxes page.

f. From the table showing the available sandboxes in the Manage Sandboxes page, select the newly created sandbox that you want to activate.

g. On the toolbar, click **Activate Sandbox**.
   The sandbox is activated.

3. Create a new UI form as follows. For detailed instructions, see the "Managing Forms" chapter in the **Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager**.
   a. In the left pane, under Configuration, click **Form Designer**. The Form Designer page is displayed.
   b. From the Actions menu, select **Create**. Alternatively, click **Create** on the toolbar. The Create Form page is displayed.
   c. On the Create Form page, enter values for the following UI fields:
      - **Resource Type**: Select the resource object that you want to associate the form with.
      - **Form Name**: Enter a name for the form.
   d. Click **Create**.
      A message is displayed stating that the form is created.

4. Create an application instance as follows. For detailed instructions, see the "Managing Application Instances" chapter in the **Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager**.
   a. In the left pane, under Configuration, click **Application Instances**. The Application Instances page is displayed.
   b. From the Actions menu, select **Create**. Alternatively, click **Create** on the toolbar. The Create Application Instance page is displayed.
   c. 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).

  d. Click Save. The application instance is created.

  e. Publish the application instance to an organization to make the application instance available for requesting and subsequent provisioning to users. See the "Managing Organizations Associated With Application Instances" section in Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager for detailed instructions.

5. Publish the sandbox as follows. For detailed instructions, see the "Managing Sandboxes" section in the Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager.

  a. In the upper right corner of the page, click the **Sandbox** link.

      The Manage Sandboxes page is displayed.

  b. From the table showing the available sandboxes, select the sandbox that you want to publish.

  c. On the toolbar, click **Publish Sandbox**.

      A message is displayed asking for confirmation.

  d. Click **Yes** to confirm. The sandbox is published.

6. Harvest entitlements and sync catalog as follows:

  a. 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 Section 4.4.1, "Scheduled Jobs for Lookup Field Synchronization for Oracle Database."

      **For MSSQL**: Run the scheduled jobs listed in Section 3.4.1, "Scheduled Jobs for Lookup Field Synchronization for MSSQL."

      **For MySQL**: Run the scheduled jobs listed in Section 5.4.1, "Scheduled Job for Lookup Field Synchronization for MySQL."

      **For DB2**: Run the scheduled jobs listed in Section 6.4.1, "Scheduled Jobs for Lookup Field Synchronization for DB2."

      **For Sybase**: Run the scheduled jobs listed in Section 7.4.1, "Scheduled Jobs for Lookup Field Synchronization for Sybase."

      **For a JDBC-Based Database**: Run the scheduled jobs listed in Section 8.10.1, "Configuring Scheduled Jobs for Lookup Field Synchronization."
b. Run the Entitlement List scheduled job to populate Entitlement Assignment schema from child process form table. See the "Predefined Scheduled Tasks" section in the Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager for more information about this scheduled job.

c. Run the Catalog Synchronization Job scheduled job. See the "Predefined Scheduled Tasks" section in the Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager for more information about this scheduled job.

2.3.1.8 Localizing Field Labels in UI Forms

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="${adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceBundle']['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.useR0.UD_DB_ORA_U_USERNAME__c_description']}">
   ```
d. Open the resource file from the connector package, for example DB-UM_ja.properties, and get the value of the attribute from the file, for example, global.udf.UD_DB_ORA_U_USERNAME=$\{adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceBundle']['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.user.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.OracleDBForm.UD_DB_ORA_U_USERNAME__c_LABEL">
<source>Username</source>
</target>
</trans-unit>

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

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

<trans-unit id="sessiondef.oracle.iam.ui.runtime.form.model.OracleDBForm.entity.OracleDBForm.UD_DB_ORA_U_USERNAME__c_LABEL">
<source>Username</source>
<target>$\{adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceBundle']['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.user.UD_DB_ORA_U_USERNAME__c_description']\}$</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: The "Deploying and Undeploying Customizations" chapter in the Oracle Fusion Middleware Developer's Guide 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:
  - Section 4.3.2, "Lookup Definitions for Configurations for Oracle Database"
  - Section 3.3.2, "Lookup Definitions for Configurations for MSSQL"
  - Section 5.3.2, "Lookup Definitions for Configurations for MySQL"
  - Section 6.3.2, "Lookup Definitions for Configurations for DB2"
  - Section 7.3.2, "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. Depending on the Oracle Identity Manager release you are using, perform one of the following steps:
   - 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 Configuration region, click **Manage IT Resource**.
   - For 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 the "Managing Sandboxes" section of Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager.
     c. In the left pane, under Configuration, click **IT Resource**.

2. 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**
3. Click the edit icon for the IT resource.
4. From the list at the top of the page, select **Details and Parameters**.
5. Specify values for the parameters of the IT resource. **Table 2-4** describes each parameter.

### Table 2-4 IT Resource Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Configuration Lookup      | This parameter holds the name of the lookup definition that stores configuration information for connector operations. If you have configured your target system as a target resource, then enter one of the following values:  
  - For Oracle: Lookup.DBUM.Oracle.Configuration  
  - For MSSQL: Lookup.DBUM.MSSQL.Configuration  
  - For MySQL: Lookup.DBUM.MySQL.Configuration  
  - For DB2: Lookup.DBUM.DB2.Configuration  
  - For Sybase: Lookup.DBUM.Sybase.Configuration  
  If you have configured your target system as a trusted source, then enter one of the following values:  
  - 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, create a new IT resource such as Lookup.DBUM.Sybase.Configuration.Trusted |
| Connector Server Name     | Specify the name of the connector server IT resource. Sample value: DBUM Connector Server |
| Connection Properties     | Specify the connection properties for the target system database.            |
| Database Name             | This parameter specifies the database name for the SQL server. Sample value: Master |
| DB Type                   | This field identifies database type (such as Oracle and MSSQL) and its used for loading respective scripts. Sample value: Oracle |
| JDBC Driver               | Depending on the target system that you are using, enter one of the following values as the JDBC driver class name:  
  - For Oracle: oracle.jdbc.driver.OracleDriver  
  - For MSSQL: microsoft.sqlserver.jdbc.SQLServerDriver  
  - For MySQL: com.mysql.jdbc.Driver  
  - For DB2: com.ibm.db2.jcc.DB2Driver  
  - For Sybase: com.sybase.jdbc4.jdbc.SybDriver |
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:

To meet the requirement posed by such a scenario:

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.

---

### Table 2–4 (Cont.) IT Resource Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JDBC URL</td>
<td>Specify the JDBC URL for the target system database.</td>
</tr>
<tr>
<td></td>
<td><strong>Sample Value:</strong></td>
</tr>
<tr>
<td></td>
<td>- For Oracle: <code>jdbc:oracle:thin:@host:port:sid</code></td>
</tr>
<tr>
<td></td>
<td>If you are using Oracle 12c PDB Mode, specify <code>jdbc:oracle:thin:@//host:port/servicename</code> as the JDBC URL.</td>
</tr>
<tr>
<td></td>
<td>- For MSSQL: <code>jdbc:sqlserver://host:port</code></td>
</tr>
<tr>
<td></td>
<td>- For MySQL: <code>jdbc:mysql://host:port/database</code></td>
</tr>
<tr>
<td></td>
<td>- For DB2: <code>jdbc:db2://server:port/databaseName</code></td>
</tr>
<tr>
<td></td>
<td>- For Sybase: <code>jdbc:sybase:Tds:host:port/database</code></td>
</tr>
<tr>
<td>Login Password</td>
<td>Enter the password for the user name of the target system account to be used for connector operations.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> 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 Section 2.3.1.4, &quot;Creating the Administrator Account on Oracle Database Vault&quot;.</td>
</tr>
<tr>
<td>Login User</td>
<td>Enter the user name of the target system account to be used for connector operations.</td>
</tr>
<tr>
<td></td>
<td>- For Oracle: <code>sys as sysdba</code></td>
</tr>
<tr>
<td></td>
<td>- For MSSQL: <code>sa</code></td>
</tr>
<tr>
<td></td>
<td>- For MySQL: <code>root</code></td>
</tr>
<tr>
<td></td>
<td>- For DB2: <code>db2admin</code></td>
</tr>
<tr>
<td></td>
<td>- For Sybase: <code>sa</code></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>- 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 Section 2.3.1.4, &quot;Creating the Administrator Account on Oracle Database Vault&quot;.</td>
</tr>
<tr>
<td></td>
<td>- The MySQL user must have all the privileges on all the schemas.</td>
</tr>
<tr>
<td></td>
<td>To grant these privileges, open the MySQL workbench, click <strong>Administration</strong> and <strong>Users and Privileges</strong>. Then, click <strong>Schema Privileges</strong>, enter <code>*</code> (the star wild card), and select all the privileges. Save the updates.</td>
</tr>
</tbody>
</table>
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:

\texttt{OIM\_HOME/server/bin/UploadJars.bat}

For UNIX:

\texttt{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.

\textbf{See Also:} Oracle Fusion Middleware Developer’s Guide 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.

\subsection*{2.3.4 Configuring the IT Resource for the Connector Server}

\textbf{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. Depending on the Oracle Identity Manager release you are using, perform one of the following steps:

   - For 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 \texttt{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 \texttt{Manage IT Resource}.

   - For 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 the "Managing Sandboxes" section of Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager.
c. In the left pane, under Configuration, click **IT Resource**.

2. In the IT Resource Name field on the Manage IT Resource page, enter **DB2® 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**

3. Click the edit icon corresponding to the Connector Server IT resource.

4. From the list at the top of the page, select **Details and Parameters**.

5. 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**

   Table 2–5 provides information about the parameters of the IT resource.
6. 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:

1. Go to `CONNECTOR_SERVER_HOME/Conf`.
2. Open the logging.properties file.
3. Edit the following entry:
   
   `.level=INFO`

4. Save and close the file.
5. Restart the connector server.

---

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Enter the host name or IP address of the computer hosting the Connector Server. Sample value: <code>HostName</code></td>
</tr>
<tr>
<td>Key</td>
<td>Enter the key for the Connector Server.</td>
</tr>
<tr>
<td>Port</td>
<td>Enter the number of the port at which the Connector Server is listening. Default value: 8763</td>
</tr>
<tr>
<td>Timeout</td>
<td>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: 0 (recommended value)</td>
</tr>
<tr>
<td>UseSSL</td>
<td>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></td>
</tr>
</tbody>
</table>

**See Also:** Section 4.1, "Configuring Secure Communication Between Oracle Database and Oracle Identity Manager" and Section 3.1, "Configuring Secure Communication Between MSSQL and Oracle Identity Manager" for information about enabling SSL.
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:** The "Managing Connector Lifecycle" chapter of Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager for detailed information of these steps

The following sections discuss the procedure to upgrade the connector:

- Section 2.4.1, "Preupgrade Steps"
- Section 2.4.2, "Upgrade Steps"
- Section 2.4.3, "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:

1. 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.
   - Production Environment
     Perform the upgrade procedure by using the silent 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".
2.4.3 Postupgrade Steps

Perform the following procedure:

1. Upload new connector jars:
   a. Use \$ORACLE_HOME/bin/UploadJars.sh utility for uploading connector jars.
   b. Upload bundle/org.identityconnectors.dbum-1.0.1116.jar as ICFBundle:
      - If thirdparty jar has to be added.
      - Move 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"

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
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.

See Also: The "Using the Form Version Control Utility" chapter of Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager for detailed information about the FVC utility

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 Section 2.3.1.7, "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 Section 2.3.1.7, "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 Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for more information.

5. Configure the upgraded IT resource of the source connector. See Section 2.3.2, "Configuring the IT Resource for the Target System" for information about configuring the IT resource.

6. Deploy the Connector Server. See Section 2.2.2.1, "Installing and Configuring the Connector Server" and Section 2.2.2.3, "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 Section 2.4.1, "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 Section 4.5, "Reconciliation from Oracle Database" and Section 3.5, "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 DBUM 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:** The "Managing Connector Lifecycle" chapter of *Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager* for detailed information about cloning connectors and the steps mentioned in this section

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.

### 2.5.1 Postcloning Configuration for User Accounts

Postcloning steps are performed mandatorily 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

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Map to</th>
<th>Qualifier</th>
<th>Literal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter return value</td>
<td>Object</td>
<td>Response Code</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>objectType</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>User</td>
</tr>
<tr>
<td>itResourceName [Cloned IT Resource Name]</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_ORA_UTRES1</td>
</tr>
<tr>
<td>childTableName[Cloned Child Form]</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_ORA_R1</td>
</tr>
</tbody>
</table>

**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 MySQL 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.
This chapter contains the following topics:

- Section 3.1, "Configuring Secure Communication Between MSSQL and Oracle Identity Manager"
- Section 3.2, "Determining Values for the JDBC URL and Connection Properties Parameters for MSSQL"
- Section 3.3, "Lookup Definitions for MSSQL"
- Section 3.4, "Scheduled Jobs for MSSQL"
- Section 3.5, "Reconciliation from MSSQL"
- Section 3.6, "Provisioning for MSSQL"
- Section 3.7, "Extending the Connector for MSSQL"

### 3.1 Configuring Secure Communication Between MSSQL and Oracle Identity Manager

**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.

**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 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 -trustcarts -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>
<thead>
<tr>
<th>Application Server</th>
<th>Truststore Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle WebLogic Server</td>
<td>If you are using Oracle jrockit_R27.3.1-jdk, then import the certificate into the</td>
</tr>
<tr>
<td></td>
<td>keystore in the following directory:</td>
</tr>
<tr>
<td></td>
<td>JROCKIT_HOME/jre/lib/security</td>
</tr>
<tr>
<td></td>
<td>If you are using the default Oracle WebLogic Server JDK, then import the certificate</td>
</tr>
<tr>
<td></td>
<td>into the keystore in following directory:</td>
</tr>
<tr>
<td></td>
<td>WEBLOGIC_HOME/jre/lib/security/cacerts</td>
</tr>
<tr>
<td></td>
<td>If you are using a JDK other than Oracle jrockit_R27.3.1-jdk or Oracle WebLogic</td>
</tr>
<tr>
<td></td>
<td>Server JDK, then import the certificate into your keystore at the following directory:</td>
</tr>
<tr>
<td></td>
<td>JAVA_HOME/jre/lib/security/cacerts</td>
</tr>
</tbody>
</table>

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 Section 2.3.4, "Configuring the IT Resource for the Connector Server."

### 3.2 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
Section 2.3.2, "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:
Determining Values for the JDBC URL and Connection Properties Parameters for MSSQL

**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.

  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.

  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

Lookup definitions used during connector operations can be categorized as follows:

- Section 3.3.1, "Lookup Definitions Synchronized with MSSQL"
- Section 3.3.2, "Lookup Definitions for Configurations for MSSQL"
- Section 3.3.3, "Lookup Definitions for Attribute Mappings for MSSQL Login Entity"
- Section 3.3.4, "Lookup Definitions for Attribute Mappings for MSSQL User Entity"
- Section 3.3.5, "Lookup Definitions for Exclusion Lists for MSSQL"
- Section 3.3.6, "Lookup Definitions for Transformation of Data in MSSQL"
- Section 3.3.7, "Lookup Definition for Validation of Data in MSSQL"

You must provide Decode values for some of the entries of the following lookup definitions. 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 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.

While performing a provisioning operation on the Administrative and User Console, 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>
<thead>
<tr>
<th>Lookup Definition</th>
<th>Target Column Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup.DBUM.MSSQL.AuthType</td>
<td>authType</td>
</tr>
<tr>
<td>Lookup.DBUM.MSSQL.DBNames</td>
<td>name</td>
</tr>
<tr>
<td>Lookup.DBUM.MSSQL.DBRoles</td>
<td>name</td>
</tr>
<tr>
<td>Lookup.DBUM.MSSQL.DefaultLang</td>
<td>alias</td>
</tr>
</tbody>
</table>

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
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:

- **Section 3.3.2.1, "Lookup.DBUM.MSSQL.Configuration"
- **Section 3.3.2.2, "Lookup.DBUM.MSSQL.Login.Configuration"
- **Section 3.3.2.3, "Lookup.DBUM.MSSQL.UM.Configuration"
- **Section 3.3.2.4, "Lookup.DBUM.MSSQL.Configuration.Trusted"
- **Section 3.3.2.5, "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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bundle Name</td>
<td>org.identityconnectors.dbum</td>
<td>This entry holds the name of the connector bundle package. Do not modify this entry.</td>
</tr>
<tr>
<td>Bundle Version</td>
<td>1.0.1116</td>
<td>This entry holds the version of the connector bundle class. Do not modify this entry.</td>
</tr>
<tr>
<td>Connector Name</td>
<td>org.identityconnectors.dbum.DBUMConnector</td>
<td>This entry holds the name of the connector class. Do not modify this entry.</td>
</tr>
<tr>
<td>disableValuesSet</td>
<td>&quot;true&quot;</td>
<td>This entry is used internally. Do not modify this entry.</td>
</tr>
<tr>
<td>User Configuration</td>
<td>Lookup.DBUM.MSSQL.UM.Configuration</td>
<td>This entry holds the name of the lookup definition that contains user-specific configuration properties. Do not modify this entry.</td>
</tr>
<tr>
<td>USERLOGIN Configuration</td>
<td>Lookup.DBUM.MSSQL.Login.Configuration</td>
<td>This entry holds the name of the lookup definition that contains login-specific configuration properties. Do not modify this entry.</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provisioning Attribute Map</td>
<td>Lookup.DBUM.MSSQL.Login.ProvAttrMap</td>
</tr>
<tr>
<td>Provisioning Exclusion List</td>
<td>Lookup.DBUM.MSSQL.Login.ProvExclusions</td>
</tr>
<tr>
<td>Provisioning Validation Lookup</td>
<td>Lookup.DBUM.MSSQL.Login.ProvValidations</td>
</tr>
<tr>
<td>Recon Attribute Defaults</td>
<td>Lookup.DBUM.MSSQL.Login.ReconDefaults</td>
</tr>
<tr>
<td>Recon Attribute Map</td>
<td>Lookup.DBUM.MSSQL/Login.ReconAttrMap</td>
</tr>
<tr>
<td>Recon Exclusion List</td>
<td>Lookup.DBUM.MSSQL/Login.ReconExclusions</td>
</tr>
<tr>
<td>Recon Transformation Lookup</td>
<td>Lookup.DBUM.MSSQL/Login.ReconTransformation</td>
</tr>
<tr>
<td>Recon Validation Lookup</td>
<td>Lookup.DBUM.MSSQL/Login.ReconValidation</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provisioning Attribute Map</td>
<td>Lookup.DBUM.MSSQL.UM.ProvAttrMap</td>
</tr>
<tr>
<td>Provisioning Exclusion List</td>
<td>Lookup.DBUM.MSSQL.UM.ProvExclusions</td>
</tr>
<tr>
<td>Provisioning Validation Lookup</td>
<td>Lookup.DBUM.MSSQL.UM.ProvValidations</td>
</tr>
<tr>
<td>Recon Attribute Defaults</td>
<td>Lookup.DBUM.MSSQL.UM.ReconDefaults</td>
</tr>
<tr>
<td>Recon Attribute Map</td>
<td>Lookup.DBUM.MSSQL.UM.ReconAttrMap</td>
</tr>
<tr>
<td>Recon Exclusion List</td>
<td>Lookup.DBUM.MSSQL.UM.ReconExclusions</td>
</tr>
<tr>
<td>Recon Transformation Lookup</td>
<td>Lookup.DBUM.MSSQL.UM.ReconTransformations</td>
</tr>
<tr>
<td>Recon Validation Lookup</td>
<td>Lookup.DBUM.MSSQL.UM.ReconValidation</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bundle Name</td>
<td>org.identityconnectors.dbum</td>
<td>This entry holds the name of the connector bundle package. Do not modify this entry.</td>
</tr>
<tr>
<td>Bundle Version</td>
<td>1.0.1116</td>
<td>This entry holds the version of the connector bundle class. Do not modify this entry.</td>
</tr>
<tr>
<td>Connector Name</td>
<td>org.identityconnectors.dbum.DBUMConnector</td>
<td>This entry holds the name of the connector class. Do not modify this entry.</td>
</tr>
<tr>
<td>disableValuesSet</td>
<td>&quot;true&quot;</td>
<td>This entry is used internally. Do not modify this entry.</td>
</tr>
<tr>
<td>USERLOGIN</td>
<td>Lookup.DBUM.MSSQL.UUM.Configuration.Trusted</td>
<td>This entry holds the name of the lookup definition that contains login-specific configuration properties. Do not modify this entry.</td>
</tr>
</tbody>
</table>

3.3.2.5 Lookup.DBUM.MSSQL.UUM.Configuration.Trusted

The Lookup.DBUM.MSSQL.UUM.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>
<thead>
<tr>
<th>Code Key</th>
<th>Dcode Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recon Attribute Defaults</td>
<td>Lookup.DBUM.MSSQL.UUM.ReconDefaults.Trusted</td>
<td></td>
</tr>
<tr>
<td>Recon Attribute Map</td>
<td>Lookup.DBUM.MSSQL.UUM.ReconAttrMap.Trusted</td>
<td></td>
</tr>
<tr>
<td>Recon Exclusion List</td>
<td>Lookup.DBUM.MSSQL.UUM.ExclusionList.Trusted</td>
<td></td>
</tr>
<tr>
<td>Recon Transformation Lookup</td>
<td>Lookup.DBUM.MSSQL.UUM.ReconTransformations.Trusted</td>
<td></td>
</tr>
<tr>
<td>Recon Validation Lookup</td>
<td>Lookup.DBUM.MSSQL.UUM.ReconValidation.Trusted</td>
<td></td>
</tr>
</tbody>
</table>

3.3.3 Lookup Definitions for Attribute Mappings for MSSQL Login Entity

This section describes the following lookup definitions:

- Section 3.3.3.1, "Lookup.DBUM.MSSQL.Login.ProvAttrMap"
- Section 3.3.3.2, "Lookup.DBUM.MSSQL.Login.ReconAttrMap"
- Section 3.3.3.3, "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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication Type</td>
<td>authType</td>
</tr>
</tbody>
</table>
### 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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication Type</td>
<td>authType</td>
</tr>
<tr>
<td>Default Database Name[LOOKUP]</td>
<td>defaultDatabase</td>
</tr>
<tr>
<td>Default Language[LOOKUP]</td>
<td>defaultLanguage</td>
</tr>
<tr>
<td>Login Name</td>
<td><strong>NAME</strong></td>
</tr>
<tr>
<td>Password</td>
<td><strong>PASSWORD</strong></td>
</tr>
<tr>
<td>Reference ID</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>Status</td>
<td><strong>ENABLE</strong></td>
</tr>
</tbody>
</table>

### 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:

- Section 3.3.4.1, "Lookup.DBUM.MSSQL.UM.ProvAttrMap"
- Section 3.3.4.2, "Lookup.DBUM.MSSQL.UM.ReconAttrMap"
- Section 3.3.4.3, "Lookup.DBUM.MSSQL.UM.ReconDefaults"
- Section 3.3.4.4, "Lookup.DBUM.MSSQL.UM.ReconAttrMap.Trusted"
- Section 3.3.4.5, "Lookup.DBUM.MSSQL.UM.ReconDefaults.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 3–10 Entries in Lookup.DBUM.MSSQL/Login.ReconAttrMap**

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default DataBase[LOOKUP]</td>
<td>defaultDatabase</td>
</tr>
<tr>
<td>Default Language[LOOKUP]</td>
<td>defaultLanguage</td>
</tr>
<tr>
<td>Login Name</td>
<td><strong>NAME</strong></td>
</tr>
<tr>
<td>Reference ID</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>Status</td>
<td><strong>ENABLE</strong></td>
</tr>
</tbody>
</table>
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
<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Name[LOOKUP]</td>
<td>defaultDatabase</td>
</tr>
<tr>
<td>Login Name</td>
<td>loginName</td>
</tr>
<tr>
<td>Reference ID</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>Role List~Role[LOOKUP]</td>
<td>roles<del>DBRole</del><strong>NAME</strong></td>
</tr>
<tr>
<td>Username</td>
<td><strong>UID</strong></td>
</tr>
</tbody>
</table>

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
<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Name</td>
<td>WITHOUT LOGIN</td>
</tr>
</tbody>
</table>

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
<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empl Type</td>
<td>Full-Time</td>
</tr>
<tr>
<td>Organization Name</td>
<td>Xellerate Users</td>
</tr>
<tr>
<td>Status</td>
<td>Active</td>
</tr>
<tr>
<td>User Type</td>
<td>End-User</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>Status[TRUSTED]</td>
<td><strong>ENABLE</strong></td>
</tr>
<tr>
<td>User ID</td>
<td><strong>UID</strong></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Sample Values</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Name</td>
<td>User ID of a user</td>
<td>Code Key: Login Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Decode: User001</td>
</tr>
<tr>
<td>Login Name with the [PATTERN] suffix</td>
<td>A regular expression supported by the representation in the java.util.regex.Pattern class</td>
<td>Code Key: Login Name[PATTERN]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>To exclude users matching any of the user ID’s User001, User002, User088, then:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Decode: User001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>To exclude users whose user ID’s start with 00012, then:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Decode: 00012*</td>
</tr>
</tbody>
</table>

See Also: For information about the supported patterns, visit [http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html](http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html)

Section 3.7.7, "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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication Type</td>
<td>mssql.oimcp.dbum.transformatiions.AuthTypeTransformationImpl</td>
</tr>
</tbody>
</table>

- 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

Section 3.7.6, “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

Section 3.7.5, “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:

- Section 3.4.1, “Scheduled Jobs for Lookup Field Synchronization for MSSQL”
- Section 3.4.2, “Attributes for Scheduled Jobs for MSSQL”
- Section 3.4.3, “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>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Key Attribute</td>
<td>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).</td>
</tr>
<tr>
<td></td>
<td>Sample value: <strong>NAME</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Do not change the value of this attribute.</td>
</tr>
<tr>
<td>Decode Attribute</td>
<td>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).</td>
</tr>
<tr>
<td></td>
<td>Sample value: <strong>NAME</strong></td>
</tr>
<tr>
<td>IT Resource Name</td>
<td>Enter the name of the IT resource for the target system installation from which you want to reconcile user records.</td>
</tr>
<tr>
<td></td>
<td>Default value: MSSQL DB</td>
</tr>
<tr>
<td>Lookup Name</td>
<td>This attribute holds the name of the lookup definition that maps each lookup definition with the data source from which values must be fetched.</td>
</tr>
<tr>
<td></td>
<td>Depending on the scheduled job you are using, the default values are as follows:</td>
</tr>
<tr>
<td></td>
<td>- For DBUM MSSQL DB Names Lookup Reconciliation - Lookup.DBUM.MSSQL.DBNames</td>
</tr>
<tr>
<td></td>
<td>- For DBUM MSSQL Languages Lookup Reconciliation - Lookup.DBUM.MSSQL.DefaultLang</td>
</tr>
<tr>
<td></td>
<td>- For DBUM MSSQL Roles Lookup Reconciliation - Lookup.DBUM.MSSQL.DBRoles</td>
</tr>
<tr>
<td>Object Type</td>
<td>Enter the type of object whose values must be synchronized.</td>
</tr>
<tr>
<td></td>
<td>Depending on the scheduled job you are using, the default values are as follows:</td>
</tr>
<tr>
<td></td>
<td>- For DBUM MSSQL DB Names Lookup Reconciliation - <strong>DBNAMES</strong></td>
</tr>
<tr>
<td></td>
<td>- For DBUM MSSQL Languages Lookup Reconciliation - <strong>LANGUAGES</strong></td>
</tr>
<tr>
<td></td>
<td>- For DBUM MSSQL Roles Lookup Reconciliation - <strong>ROLES</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Do not change the value of this attribute.</td>
</tr>
<tr>
<td>Resource Object Name</td>
<td>Enter the name of the resource object that is used for reconciliation.</td>
</tr>
<tr>
<td></td>
<td>Default value: MSSQL DB User</td>
</tr>
</tbody>
</table>

### 3.3.2 Attributes for Scheduled Jobs for MSSQL

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
- DBUM MSSQL Delete User Target Reconciliation
For MSSQL login entity:

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Expression for filtering records that must be reconciled by the scheduled job</td>
</tr>
<tr>
<td></td>
<td>By default, the value of this attribute is empty.</td>
</tr>
<tr>
<td></td>
<td>Sample value: equalTo('<strong>Name</strong>', 'SEPT12USER1')</td>
</tr>
<tr>
<td></td>
<td>See Section 3.5.9, &quot;Performing Limited Reconciliation from MSSQL&quot; for the syntax of this expression.</td>
</tr>
<tr>
<td>Incremental Recon Attribute</td>
<td>Time stamp at which the last reconciliation run started</td>
</tr>
<tr>
<td></td>
<td>Sample value: lastModified</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Do not enter a value for this attribute. The reconciliation engine automatically enters a value for this attribute.</td>
</tr>
<tr>
<td>IT Resource Name</td>
<td>Name of the IT resource for the target system installation from which you want to reconcile user records</td>
</tr>
<tr>
<td></td>
<td>Default value: MSSQL DB</td>
</tr>
<tr>
<td>Latest Token</td>
<td>This attribute is used for internal purposes.</td>
</tr>
<tr>
<td></td>
<td>By default, this value is empty.</td>
</tr>
<tr>
<td>Object Type</td>
<td>Type of object you want to reconcile</td>
</tr>
<tr>
<td></td>
<td>For trusted mode and login entity scheduled jobs: USERLOGIN</td>
</tr>
<tr>
<td></td>
<td>Default value: User</td>
</tr>
<tr>
<td>Resource Object Name</td>
<td>Name of the resource object that is used for reconciliation</td>
</tr>
<tr>
<td></td>
<td>For login entity scheduled jobs: MSSQL DB User Login</td>
</tr>
<tr>
<td></td>
<td>For trusted mode scheduled jobs: MSSQL UserLogin Trusted</td>
</tr>
<tr>
<td></td>
<td>For all other scheduled jobs: MSSQL DB User</td>
</tr>
<tr>
<td>Scheduled Task Name</td>
<td>Name of the scheduled job</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> 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.</td>
</tr>
</tbody>
</table>

Table 3–19 describes the attributes of the scheduled jobs for delete operations.
### Table 3–19 Attributes of the Scheduled Jobs for Delete Operations

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

### 3.4.3 Configuring Scheduled Jobs for MSSQL

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

See Section 3.4.1, "Scheduled Jobs for Lookup Field Synchronization for MSSQL" and Section 3.4.2, "Attributes for Scheduled Jobs for MSSQL" for the scheduled jobs that are part of the connector and for information about their attributes.

To configure a scheduled job:

1. Depending on the Oracle Identity Manager release you are using, perform one of the following steps:
   - 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.
   - For Oracle Identity Manager release 11.2.x or later:
     a. Log in to Oracle Identity System Administration.
     b. In the left pane, under System Management, click Scheduler.

2. Search for and open the scheduled job as follows:
   - 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.
   - In the search results table on the left pane, click the scheduled job in the Job Name column.

3. 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 Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager for detailed information about schedule types.

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

4. 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 Section 3.4.2, "Attributes for Scheduled Jobs for MSSQL."

5. After specifying the attributes, click Apply to save the changes.

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

3.5 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.

- Section 3.5.1, "Guidelines on Configuring Reconciliation from MSSQL"
- Section 3.5.2, "Reconciliation Process for MSSQL"
- Section 3.5.3, "Reconciliation Queries for MSSQL"
- Section 3.5.4, "Target System Columns Used in Reconciliation from MSSQL"
- Section 3.5.5, "Configuring the Target System As a Trusted Source"
- Section 3.5.6, "Reconciliation Rules for MSSQL"
- Section 3.5.7, "Reconciliation Action Rules for MSSQL"
- Section 3.5.8, "Performing Full Reconciliation from MSSQL"
- Section 3.5.9, "Performing Limited 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 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 Section 3.5.5, "Configuring the Target System As a Trusted Source" for more information.

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:
   a. Each user record fetched from the target system is compared with existing OIM Users. The reconciliation rule is applied during the comparison process. See Section 3.5.6, “Reconciliation Rules for MSSQL” for information about the reconciliation rule.
   b. 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:
   a. 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. See Section 3.5.6, "Reconciliation Rules for MSSQL" for information about the reconciliation rule.
   b. 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.

3.5.3 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. Section 3.7.1, "Guidelines on Configuring the 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

---

Reconciliation from MSSQL
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 Section 3.3.3, "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 Section 3.3.4, "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. Depending on the Oracle Identity Manager release you are using, perform one of the following steps:
   - For Oracle Identity Manager release 11.1.1.x:
     Log in to the Administrative and User Console
   - For Oracle Identity Manager release 11.1.2.x or later:
     Log in to Oracle Identity System Administration

2. If you are using Oracle Identity Manager release 11.1.1.x, then:
   a. On the Welcome page, click **Advanced** in the upper-right corner of the page.
   b. On the Welcome to Oracle Identity Manager Advanced Administration page, in the Configuration region, click **Create IT Resource**.

3. If you are using Oracle Identity Manager release 11.1.2.x or later, then:
   a. In the left pane, under Configuration, click **IT Resource**.
   b. In the Manage IT Resource page, click **Create IT Resource**.
4. 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.

5. Click **Continue**.

   - **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.

7. 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**: Oracle Fusion Middleware User’s Guide for 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.
After you deploy the connector, you can view the reconciliation rule for reconciliation by performing the following steps:

1. Log in to the Oracle Identity Manager Design Console.
2. Expand Development Tools.
4. Search for the rule name.

**3.5.7 Reconciliation Action Rules for MSSQL**

This section provides information on the reconciliation action rules for reconciliation.

No action is performed for rule conditions that are not predefined for this connector. You can define your own action rule for such rule conditions. See Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for information about modifying or creating reconciliation action rules.

Table 3–20 lists the action rules for target resource reconciliation.

<table>
<thead>
<tr>
<th>Rule Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Matches Found</td>
<td>Assign to Administrator With Least Load</td>
</tr>
<tr>
<td>One Entity Match Found</td>
<td>Establish Link</td>
</tr>
<tr>
<td>One Process Match Found</td>
<td>Establish Link</td>
</tr>
</tbody>
</table>

Table 3–21 lists the action rules for trusted source reconciliation.

<table>
<thead>
<tr>
<th>Rule Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Matches Found</td>
<td>Create User</td>
</tr>
<tr>
<td>One Entity Match Found</td>
<td>Establish Link</td>
</tr>
</tbody>
</table>

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

Table 3–21 Action Rules for Trusted Source Reconciliation 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.
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
3.5.8 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 Section 3.4.2, “Attributes for Scheduled Jobs for MSSQL” for information about this scheduled job.

3.5.9 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 the "ICF Filter Syntax" section of the Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager.

While deploying the connector, follow the instructions in Section 3.4.3, “Configuring Scheduled Jobs for MSSQL” to specify attribute values.

3.5.10 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.

**See Also:** The "Provisioning" section in Oracle Fusion Middleware User’s Guide for Identity Manager for conceptual information about provisioning

This section contains the following topics about provisioning:

- Section 3.6.1, "Guidelines on Performing Provisioning Operations for MSSQL"
- Section 3.6.2, "Provisioning Process for MSSQL"
- Section 3.6.3, "Configuring Direct Provisioning for MSSQL"
- Section 3.6.4, "Configuring Request-Based Provisioning for MSSQL"
- Section 3.6.5, "Switching Between Request-Based Provisioning and Direct Provisioning for MSSQL"
- Section 3.6.6, "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 Provisioning Process for MSSQL

**See Also:** The “Provisioning” section in Oracle Fusion Middleware User’s Guide for Identity Manager for conceptual information about provisioning

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.

This 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 Section 3.6.5, "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:** sqldb1
- **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

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.

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

- Section 3.6.4.1, "Approver’s Role in Request-Based Provisioning for MSSQL"
- Section 3.6.4.2, "Importing MSSQL Request Datasets Using Deployment Manager"
- Section 3.6.4.3, "End User’s Role in Request-Based Provisioning for MSSQL"
- Section 3.6.4.4, "Enabling the Auto Save Form Feature for MSSQL"
- Section 3.6.4.5, "Running the PurgeCache Utility for MSSQL"

3.6.4.1 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 Importing MSSQL Request Datasets Using Deployment Manager

See Also: Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager for detailed information about importing objects from an XML file using the 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.

3.6.4.3 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:

See Also: Oracle Fusion Middleware User’s Guide for Identity Manager for detailed information about these steps

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
   - 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.

### 3.6.4.4 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.

### 3.6.4.5 Running the PurgeCache Utility for MSSQL

Run the PurgeCache utility to clear content belonging to the Metadata category from the server cache. See Section 2.3.1.3, "Clearing Content Related to 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

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 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.

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 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.

Note: It is assumed that you have performed the procedure described in Section 3.6.4, "Configuring Request-Based Provisioning for MSSQL."
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 Oracle Identity Administrative and User console.
2. If you want to first create an OIM User and then provision a target system account, then:

   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:

   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:
3.7.1 Guidelines on Configuring 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.
The following sections discuss guidelines that you must apply while modifying the predefined queries or creating new queries:

- Section 3.7.1.1, "Syntax of Provisioning Queries for MSSQL"
- Section 3.7.1.2, "Syntax of Reconciliation Queries for MSSQL"
- Section 3.7.1.3, "Syntax of List of Values Queries for MSSQL"
- Section 3.7.1.4, "Guidelines for Configuring Search Queries Used in Reconciliation from MSSQL"

3.7.1.1 Syntax of Provisioning Queries for MSSQL
The following is the syntax of the queries used for provisioning operations:

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

For example:

```plaintext
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>
<thead>
<tr>
<th>Operation</th>
<th>Selection Logic</th>
<th>Query IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>CREATE_AUHTTYPE_OBJECTYPE CREATE_OBJECTTYPE</td>
<td>CREATE_SQL_SERVER_AUTHENTICATION_USERLOGIN CREATE_WINDOWS_AUTHENTICATION_USERLOGIN CREATE_USER</td>
</tr>
<tr>
<td>DELETE</td>
<td>DELETE_OBJECTTYPE</td>
<td>DELETE_USERLOGIN DELETE_USER</td>
</tr>
</tbody>
</table>
3.7.1.2 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":"PARAM_DEF_N1", "PARAM2":"PARAM_DEF_N2"...

ExtensionJoin="EXTENSIONJOIN"

ExtensionSeparator="EXTENSION_SEPARATOR"

QueryExtensions="EXTENSION1","EXTENSION2"...

For example:

```sql
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**
- **ENABLE**
- **DISABLE**
- **RESET PASSWORD**
- **UPDATE**
- **ADD CHILD VALUES**
- **REMOVE CHILD VALUES**
- **ENABLE OBJECTTYPE**
- **DISABLE OBJECTTYPE**
- **USEROBJECT_SET_PASSWORD**
- **USERLOGIN_SET_PASSWORD**
- **USERLOGIN_UPDATE_DEFAULTDATABASE**
- **USERLOGIN_UPDATE_DEFAULTLANGUAGE**
- **USERLOGIN_UPDATE_LOGINNAME**
- **USERDATA_QUERY**
- **UPDATE_ADD_ATTRIBUTE**
- **UPDATE_ADD_ROLES**
- **UPDATE_REVOKE_ATTRIBUTE**
- **UPDATE_REVOKE_ROLES**

### Table 3-22 (Cont.) Script Section Logic for MSSQL Provisioning Queries

<table>
<thead>
<tr>
<th>Operation</th>
<th>Selection Logic</th>
<th>Query IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENABLE</td>
<td>ENABLE_OBJECTTYPE</td>
<td>ENABLE_USERLOGIN</td>
</tr>
<tr>
<td>DISABLE</td>
<td>DISABLE_OBJECTTYPE</td>
<td>DISABLE_USERLOGIN</td>
</tr>
<tr>
<td>RESET PASSWORD</td>
<td>OBJECTTYPE_SET_PASSWORD</td>
<td>USERLOGIN_SET_PASSWORD</td>
</tr>
<tr>
<td>UPDATE</td>
<td>OBJECTTYPE_UPDATE_ATTRIBUTE</td>
<td>USERLOGIN_UPDATE_DEFAULTDATABASE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USERLOGIN_UPDATE_DEFAULTLANGUAGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USERLOGIN_UPDATE_LOGINNAME</td>
</tr>
<tr>
<td>ADD CHILD VALUES</td>
<td>UPDATE_ADD_ATTRIBUTE</td>
<td>UPDATE_ADD_ROLES</td>
</tr>
<tr>
<td>REMOVE CHILD VALUES</td>
<td>UPDATE_REVOKE_ATTRIBUTE</td>
<td>UPDATE_REVOKE_ROLES</td>
</tr>
</tbody>
</table>
Query refers to the main query, represented by `QUERY`.
For example: `Query="select {__UID__}, {lastModified} from (select * from sys.sysusers where issqrlrole=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 `QUERY_TYPE` can be `SQL`, `StoredProc`, or `QUERY_EXTENSION`.
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 `EXTENSION_SEPARATOR`.
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 EXTENSION_SEPARATOR EXTENSION2 EXTENSION_SEPARATOR...]]
```

For example:

```
SELECT {__UID__}, {lastModified} FROM sys.sysusers, USER_DATA_QUERY
```
3.7.1.3 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:

\[ \text{OBJECTTYPE} = "\text{QUERY}" \]

For example:

\[ \text{__DBNAMES__} = \text{"SELECT name FROM sys.sysdatabases"} \]

In this syntax:

- \text{OBJECTTYPE} refers to the attribute provided by objectType scheduled job parameter.
  
  For example: \text{__DBNAMES__}

- \text{QUERY} refers to the query used for fetching a lookup field attribute.
  
  For example: \text{SELECT name FROM sys.sysdatabases}

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

3.7.1.4 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)"} 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.
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:

- Section 3.7.2.1, "Updating a Query File for MSSQL"
- Section 3.7.2.2, "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.

   See Also: Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for detailed information about the Download JARs utility

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`
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 Section 3.7.1, "Guidelines on Configuring the 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.

```
See Also: Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for detailed information about the Update JARs utility
```

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. See Step 2 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for more information.
   c. Create a new UI form to view the newly added field along with the rest of the fields. See Step 3 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for 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 4.c), and then save the application instance.
   e. Publish the sandbox. See Step 5 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for more information.

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:
   b. Double-click Process Definition and open the MSSQL DB User process definition.
   c. In the process definition, add a new task for updating the field as follows:
      – Click Add and enter the task name, for example, Custom Attribute 1 Updated, and the task description.
      – In the Task Properties section, select the following fields:
         Conditional
         Allow Multiple Instances
      – Click the save icon.
   d. On the Integration tab, click Add, and then click Adapter.
   e. Select the adpMSSQLUPDATEUSER 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:
Extending the Connector for MSSQL

3.7.3 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.

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.

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Map To</th>
<th>Qualifier</th>
<th>Literal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter return value</td>
<td>Object</td>
<td>Response code</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>attributeName</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>Custom Attribute 1</td>
</tr>
<tr>
<td>itRes</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_SQL_U_ITRES</td>
</tr>
<tr>
<td>objectType</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>User</td>
</tr>
<tr>
<td>processInstanceKey</td>
<td>Long</td>
<td>Process Data</td>
<td>Process Instance</td>
<td>NA</td>
</tr>
</tbody>
</table>

Table 3–23

<table>
<thead>
<tr>
<th>Code Name</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>Error occurred</td>
<td>R</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>An unknown response was received</td>
<td>R</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Operation completed</td>
<td>C</td>
</tr>
</tbody>
</table>

g. On the Responses tab, click Add to add the following response codes:

h. Click the save icon and then close the dialog box.
**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 Section 2.3.1.7, "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

<table>
<thead>
<tr>
<th>Connector Object</th>
<th>Name</th>
<th>Referenced By</th>
<th>Comments on Creating a Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT resource</td>
<td>MSSQL DB</td>
<td>Process forms:</td>
<td>You need to create a copy of IT resource with a different name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ UD_DB_SQL_L</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ UD_DB_SQL_U</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scheduled Jobs</td>
<td></td>
</tr>
<tr>
<td>Resource object</td>
<td>MSSQL DB</td>
<td>All connector operations</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>User Login</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MSSQL DB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>User</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scheduled Jobs</td>
<td>There are many scheduled</td>
<td>NA</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>jobs for different purposes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process definition</td>
<td>MSSQL DB</td>
<td>NA</td>
<td>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. Note: Create copies of the process form only if there are differences in attributes between the various installations of the target system.</td>
</tr>
<tr>
<td></td>
<td>User Login</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MSSQL DB</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>User</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process form</td>
<td>UD_DB_SQL_L</td>
<td>Process definitions:</td>
<td>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. Note: Create copies of the process form only if there are differences in attributes between the various installations of the target system.</td>
</tr>
<tr>
<td></td>
<td>UD_DB_SQL_U</td>
<td>■ MSSQL DB User Login</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ MSSQL DB User</td>
<td></td>
</tr>
</tbody>
</table>
When you configure reconciliation:

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.

<table>
<thead>
<tr>
<th>Connector Object</th>
<th>Name</th>
<th>Referred By</th>
<th>Comments on Creating a Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Child process form</td>
<td>UD_DB_SQL_R</td>
<td>MSSQL DB User (Process definition)</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UD_DB_SQL_U (Process form)</td>
<td></td>
</tr>
<tr>
<td>Configuration lookup definition for a target system configured as a target resource</td>
<td>Lookup.DBUM. MSSQL.Configuration</td>
<td>MSSQL DB (IT resource)</td>
<td>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.</td>
</tr>
<tr>
<td>Configuration lookup definition for a target system configured as a trusted source</td>
<td>Lookup.DBUM. MSSQL.Configuration.Trusted</td>
<td>MSSQL DB (IT resource)</td>
<td>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. 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.</td>
</tr>
<tr>
<td>Resource object attributes mapping lookup definition (for trusted source)</td>
<td>Lookup.DBUM. MSSQL.UM.ReconAttrMap.Trusted</td>
<td></td>
<td>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. Note: Create copies of this lookup definition only if there are differences in attributes between the two installations of the target system.</td>
</tr>
</tbody>
</table>
When you perform provisioning operations:
When you use the Administrative and User Console to perform provisioning, you can specify the IT resource corresponding to the target system installation to which you want to provision the user.

3.7.4 Configuring the Connector for Multiple Trusted Source Reconciliation from MSSQL

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.

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.
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 (#):

```java
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 Section 3.3.7, "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.

   **See Also:** *Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager* for detailed information about the Upload JARs utility

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.DBUMTransfomation`.
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:

```java
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 Section 3.3.6, "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.DBUMTransfomation.

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:

```
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.

**See Also:** *Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager* for detailed information about the Upload JARs utility

**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.

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 Section 3.3.5, "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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Name</td>
<td>User001</td>
</tr>
</tbody>
</table>

**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.

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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Name</td>
<td>User001</td>
</tr>
</tbody>
</table>

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](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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Name[PATTERN]</td>
<td>User001</td>
</tr>
</tbody>
</table>

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:
6. Click the save icon.

3.7.8 Configuring 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.

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.
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.

- **Section 4.1, "Configuring Secure Communication Between Oracle Database and Oracle Identity Manager"**
- **Section 4.2, "Determining Values for the JDBC URL and Connection Properties Parameters for Oracle Database"**
- **Section 4.3, "Lookup Definitions for Oracle Database"**
- **Section 4.4, "Scheduled Jobs for Oracle Database"**
- **Section 4.5, "Reconciliation from Oracle Database"**
- **Section 4.6, "Provisioning for Oracle Database"**
- **Section 4.7, "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:

- Section 4.1.1, "Configuring Data Encryption and Integrity in Oracle Database"
- Section 4.1.2, "Configuring SSL Communication in Oracle Database"

### 4.1.1 Configuring Data Encryption and Integrity in Oracle Database

Refer to Oracle Database Advanced Security Administrator’s Guide for information about configuring data encryption and integrity.

### 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:

   ```shell
   ..\..\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.
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 Section 2.3.4, "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 Section 2.3.2, "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:

- **Section 4.2.1, "Only Data Encryption and Integrity Is Configured"
- **Section 4.2.2, "Only SSL Communication Is Configured"
- **Section 4.2.3, "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:

---

<table>
<thead>
<tr>
<th>Application Server</th>
<th>Truststore Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle WebLogic Server</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If you are using Oracle jrockit_R27.3.1-jdk, then import the certificate into the keystore in the following directory:</td>
</tr>
<tr>
<td></td>
<td>JROCKIT_HOME/jre/lib/security</td>
</tr>
<tr>
<td></td>
<td>If you are using the default Oracle WebLogic Server JDK, then import the certificate into the keystore in following directory:</td>
</tr>
<tr>
<td></td>
<td>WEBLOGIC_HOME/java/jre/lib/security/cacerts</td>
</tr>
<tr>
<td></td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>JAVA_HOME/jre/lib/security/cacerts</td>
</tr>
</tbody>
</table>

---

Table 4–1  Truststore Locations on Supported Application Servers

<table>
<thead>
<tr>
<th>Application Server</th>
<th>Truststore Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle WebLogic Server</td>
<td>If you are using Oracle jrockit_R27.3.1-jdk, then import the certificate into the keystore in the following directory:</td>
</tr>
<tr>
<td></td>
<td>JROCKIT_HOME/jre/lib/security</td>
</tr>
<tr>
<td></td>
<td>If you are using the default Oracle WebLogic Server JDK, then import the certificate into the keystore in following directory:</td>
</tr>
<tr>
<td></td>
<td>WEBLOGIC_HOME/java/jre/lib/security/cacerts</td>
</tr>
<tr>
<td></td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>JAVA_HOME/jre/lib/security/cacerts</td>
</tr>
</tbody>
</table>
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:

```
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)))
```

See Also: Oracle Database Advanced Security Administrator’s Guide for information about the sqlnet.ora file

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.
Determining Values for the JDBC URL and Connection Properties Parameters for Oracle Database

Sample Format 3:

```plaintext
ADDRESS=(PROTOCOL_ADDRESS_INFORMATION)
CONNECT_DATA=
  (SERVICE_NAME=SERVICE_NAME))

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:

```plaintext
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:

```plaintext
javax.net.ssl.trustStore=STORE_LOCATION
javax.net.ssl.trustStoreType=JKS
javax.net.ssl.trustStorePassword=STORE_PASSWORD
```

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 elements.
Determining Values for the JDBC URL and Connection Properties Parameters for Oracle Database

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 Section 4.2.1, "Only Data Encryption and Integrity Is Configured" and Section 4.2.2, "Only SSL Communication Is Configured."

  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:

  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)))

  **Note:** The JDBC URL connection string must not exceed 200 characters.

  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= srvce1)))

- **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 Section 4.2.1, "Only Data Encryption and Integrity Is Configured" and Section 4.2.2, "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, javax.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 for Oracle Database

Lookup definitions used during connector operations can be categorized as follows:

- Section 4.3.1, "Lookup Definitions Synchronized with Oracle Database"
- Section 4.3.2, "Lookup Definitions for Configurations for Oracle Database"
- Section 4.3.3, "Lookup Definitions for Attribute Mappings for Oracle Database"
- Section 4.3.4, "Lookup Definitions for Exclusion Lists for Oracle Database"
- Section 4.3.5, "Lookup Definitions for Transformation of Data in Oracle Database"
- Section 4.3.6, "Lookup Definition for Validation of Data in Oracle Database"

You must provide Decode values for some of the entries of the following lookup definitions. 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.

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 on the Administrative and User Console, 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

<table>
<thead>
<tr>
<th>Lookup Definition</th>
<th>Target Table Name</th>
<th>Target Column Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup.DBUM.Oracle.AuthType</td>
<td>dba_users</td>
<td>DECODE(PASSWORD, 'EXTERNAL', 'EXTERNAL', 'GLOBAL', 'GLOBAL', 'PASSWORD')</td>
</tr>
<tr>
<td>Lookup.DBUM.Oracle.Privileges</td>
<td>DBA_SYS_PRIVS</td>
<td>PRIVILEGE</td>
</tr>
<tr>
<td>Lookup.DBUM.Oracle.Profiles</td>
<td>dba_users</td>
<td>DISTINCT profile</td>
</tr>
<tr>
<td>Lookup.DBUM.Oracle.Roles</td>
<td>DBA_ROLE_PRIVS</td>
<td>GRANTED_ROLE</td>
</tr>
<tr>
<td>Lookup.DBUM.Oracle.Temp.Tablespace</td>
<td>dba_users</td>
<td>DEFAULT_TABLESPACE</td>
</tr>
<tr>
<td>Lookup.DBUM.Oracle.Tablespaces</td>
<td>dba_users</td>
<td>TEMPORARY_TABLESPACE</td>
</tr>
<tr>
<td>Lookup.DBUM.Oracle.WithAdminOption</td>
<td>DBA_SYS_PRIVS, DBA_ROLE_PRIVS</td>
<td>ADMIN_OPTION</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTERNAL</td>
<td>EXTERNAL</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>GLOBAL</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>PASSWORD</td>
</tr>
</tbody>
</table>

### 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

- Section 4.3.2.1, "Lookup.DBUM.Oracle.Configuration"
- Section 4.3.2.2, "Lookup.DBUM.Oracle.UM.Configuration"
- Section 4.3.2.3, "Lookup.DBUM.Oracle.Configuration.Trusted"
- Section 4.3.2.4, "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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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 |
| reservedWordsList | "DROP","INSERT","ALTER","CREATE", "DELETE","UPDATE","GRANT","TRUNCATE", "EXEC","TEMPORARY","TABLESPACE","DEFAULT", "QUOTA","PROFILE","IDENTIFIED","EXTERNALLY", "AS","GLOBALLY","REVOKE","ACCOUNT","UNLOCK", "LOCK","CASCADE" | List of words that are reserved and are not allowed to be used in the names of the connector artifacts |

### 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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Configuration Lookup</td>
<td>Lookup.DBUM.Oracle.UM.Configuration</td>
<td>Name of the lookup definition that contains user-specific configuration properties</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Do not modify this entry.</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provisioning Attribute Map</td>
<td>Lookup.DBUM.Oracle.UM.ProvAttrMap</td>
</tr>
<tr>
<td>Provisioning Exclusion List</td>
<td>Lookup.DBUM.Oracle.UM.ExclusionList</td>
</tr>
<tr>
<td>Provisioning Validation Lookup</td>
<td>Lookup.DBUM.Oracle.UM.ProvValidations</td>
</tr>
<tr>
<td>Recon Validation Lookup</td>
<td>Lookup.DBUM.Oracle.UM.ReconValidations</td>
</tr>
<tr>
<td>Recon Attribute Map</td>
<td>Lookup.DBUM.Oracle.UM.ReconAttrMap</td>
</tr>
<tr>
<td>Recon Exclusion List</td>
<td>Lookup.DBUM.Oracle.UM.ExclusionList</td>
</tr>
<tr>
<td>Recon Transformation Lookup</td>
<td>Lookup.DBUM.Oracle.UM.ReconTransformations</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bundle Name</td>
<td>org.identityconnectors.dbum</td>
</tr>
<tr>
<td>Bundle Version</td>
<td>1.0.1116</td>
</tr>
<tr>
<td>Connector Name</td>
<td>org.identityconnectors.dbum.DBUMConnector</td>
</tr>
<tr>
<td>disableValuesSet</td>
<td>&quot;EXPIRED &amp; LOCKED&quot;,&quot;LOCKED&quot;,&quot;EXPIRED&quot;</td>
</tr>
<tr>
<td>User Configuration Lookup</td>
<td>Lookup.DBUM.Oracle.UM.Configuration.Trusted</td>
</tr>
</tbody>
</table>

4.3.3 Lookup Definitions for Attribute Mappings for Oracle Database
This section describes the following lookup definitions:

- Section 4.3.3.1, "Lookup.DBUM.Oracle.UM.ProvAttrMap"
- Section 4.3.3.2, "Lookup.DBUM.Oracle.UM.ReconAttrMap"
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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Key</td>
<td></td>
</tr>
<tr>
<td>Authentication Type</td>
<td>authType</td>
</tr>
<tr>
<td>Default Tablespace[LOOKUP]</td>
<td>tablespace</td>
</tr>
<tr>
<td>Default Tablespace Quota (in MB)</td>
<td>defaultQuota</td>
</tr>
<tr>
<td>Global DN</td>
<td>globalDN</td>
</tr>
<tr>
<td>Password</td>
<td><strong>PASSWORD</strong></td>
</tr>
<tr>
<td>Profile Name[LOOKUP]</td>
<td>profile</td>
</tr>
<tr>
<td>Return Id</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>Temporary Tablespace[LOOKUP]</td>
<td>tempTableSpace</td>
</tr>
<tr>
<td>UD_DB_ORA_P~Privilege[LOOKUP]</td>
<td>privileges<del>DBPrivilege</del><strong>NAME</strong></td>
</tr>
<tr>
<td>UD_DB_ORA_P~Privilege Admin Option</td>
<td>privileges<del>DBPrivilege</del>adminOption</td>
</tr>
<tr>
<td>UD_DB_ORA_R~Role[LOOKUP]</td>
<td>roles<del>DBRole</del><strong>NAME</strong></td>
</tr>
<tr>
<td>UD_DB_ORA_R~Role Admin Option</td>
<td>roles<del>DBRole</del>adminOption</td>
</tr>
<tr>
<td>Username</td>
<td><strong>NAME</strong></td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Key</td>
<td></td>
</tr>
<tr>
<td>Account Status</td>
<td>status</td>
</tr>
<tr>
<td>Authentication Type</td>
<td>authType</td>
</tr>
<tr>
<td>Default Tablespace[LOOKUP]</td>
<td>tablespace</td>
</tr>
<tr>
<td>Default Tablespace Quota</td>
<td>defaultQuota</td>
</tr>
<tr>
<td>Global DN</td>
<td>globalDN</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privilege List~Privilege Admin Option</td>
<td>privileges<del>DBPrivilege</del>adminOption</td>
</tr>
<tr>
<td>Privilege List~Privilege Name[LOOKUP]</td>
<td>privileges<del>DBPrivilege</del><strong>NAME</strong></td>
</tr>
<tr>
<td>Profile Name[LOOKUP]</td>
<td>profile</td>
</tr>
<tr>
<td>Reference ID</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>Role List~Role Admin Option</td>
<td>roles<del>DBRole</del>adminOption</td>
</tr>
<tr>
<td>Role List~Role Name[LOOKUP]</td>
<td>roles<del>DBRole</del><strong>NAME</strong></td>
</tr>
<tr>
<td>Status</td>
<td><strong>ENABLE</strong></td>
</tr>
<tr>
<td>Temporary Tablespace[LOOKUP]</td>
<td>tempTableSpace</td>
</tr>
<tr>
<td>User Name</td>
<td><strong>UID</strong></td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empl Type</td>
<td>Full-Time</td>
</tr>
<tr>
<td>Organization Name</td>
<td>Xellerate Users</td>
</tr>
<tr>
<td>Status</td>
<td>Active</td>
</tr>
<tr>
<td>User Type</td>
<td>End-User</td>
</tr>
</tbody>
</table>
4.3.4 Lookup Definitions for Exclusion Lists for Oracle Database

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.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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
<th>Sample Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>User ID of a user</td>
<td>Code Key: User Name</td>
</tr>
<tr>
<td></td>
<td>Decode: User001</td>
<td></td>
</tr>
<tr>
<td>User Name with the [PATTERN] suffix</td>
<td>A regular expression supported by the representation in the java.util.regex.Pattern class</td>
<td>Code Key: User Name[PATTERN]</td>
</tr>
<tr>
<td></td>
<td>To exclude users matching any of the user ID’s User001, User002, User088, then:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Decode: User001</td>
<td>User002</td>
</tr>
<tr>
<td></td>
<td>To exclude users whose user ID’s start with 00012, then:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Decode: 00012*</td>
<td></td>
</tr>
</tbody>
</table>

**See Also:** For information about the supported patterns, visit [http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html](http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html)

Section 4.7.7, “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

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.Oracle.UM.ReconTransformations

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privilege List</td>
<td>oracle.iam.connectors.dbum.transformations.OraclePrivilegeAdminOptionTransformation</td>
</tr>
<tr>
<td>Role List</td>
<td>oracle.iam.connectors.dbum.transformations.OracleRoleAdminOptionTransformation</td>
</tr>
</tbody>
</table>
Listed below are the scheduled jobs for the Oracle Database:

- For trusted source mode:
  - Lookup.DBUM.Oracle.UM.ReconTransformations.Trusted

Section 4.7.6, "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.

Section 4.7.5, "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:

- Section 4.4.1, "Scheduled Jobs for Lookup Field Synchronization for Oracle Database"
- Section 4.4.2, "Attributes for Scheduled Jobs for Oracle Database"
- Section 4.4.3, "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. The procedure to configure scheduled jobs is described later in the guide.
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
Scheduled Jobs for Oracle Database

Table 4–14 describes the attributes of the scheduled jobs for user operations.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch Size</td>
<td>Value for running the scheduled job in batch mode. By default, this value is empty.</td>
</tr>
<tr>
<td>Filter</td>
<td>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 Section 4.5.9, &quot;Performing Limited Reconciliation from Oracle Database&quot; for the syntax of this expression.</td>
</tr>
<tr>
<td>Incremental Recon Attribute</td>
<td>Time stamp at which the last reconciliation run started. Sample value: <code>lastModified</code> <strong>Note</strong>: Do not enter a value for this attribute. The reconciliation engine automatically enters a value for this attribute.</td>
</tr>
<tr>
<td>IT Resource Name</td>
<td>Name of the IT resource for the target system installation from which you want to reconcile user records. Default value: <code>Oracle DB</code></td>
</tr>
<tr>
<td>Latest Token</td>
<td>This attribute is used for internal purposes. By default, this value is empty.</td>
</tr>
<tr>
<td>Object Type</td>
<td>Type of object you want to reconcile. Default value: <code>User</code></td>
</tr>
<tr>
<td>Resource Object Name</td>
<td>Name of the resource object that is used for reconciliation. Default value: <code>Oracle DB User</code></td>
</tr>
<tr>
<td>Scheduled Task Name</td>
<td>Name of the scheduled job. <strong>Note</strong>: 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.</td>
</tr>
</tbody>
</table>

Table 4–15 describes the attributes of the scheduled jobs for delete operations.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT Resource Name</td>
<td>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.</td>
</tr>
<tr>
<td>Object Type</td>
<td>Type of object you want to reconcile. Default value: <code>User</code></td>
</tr>
<tr>
<td>Resource Object Name</td>
<td>Name of the resource object that is used for reconciliation. For DBUM Oracle Delete User Target Reconciliation: <code>Oracle DB User</code> For DBUM Oracle Delete User Trusted Reconciliation: <code>Oracle DB Trusted</code></td>
</tr>
</tbody>
</table>
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 Section 4.4.1, "Scheduled Jobs for Lookup Field Synchronization for Oracle Database" and Section 4.4.2, "Attributes for 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. Depending on the Oracle Identity Manager release you are using, perform one of the following steps:
   - 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.
   - For 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.

2. Search for and open the scheduled job as follows:
   - 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.
   - In the search results table on the left pane, click the scheduled job in the Job Name column.

3. 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 Oracle Fusion Middleware Administrator's Guide for Oracle Identity Manager for detailed information about schedule types.

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

4. On the Job Details tab, in the Parameters region, specify values for the attributes of the scheduled job.
5. After specifying the attributes, click **Apply** to save the changes.

---

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

---

### 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:

- Section 4.5.1, "Guidelines on Configuring Reconciliation for Oracle Database"
- Section 4.5.2, "Reconciliation Process for Oracle Database"
- Section 4.5.3, "Reconciliation Queries for Oracle Database"
- Section 4.5.4, "Target System Columns Used in Reconciliation from Oracle Database"
- Section 4.5.5, "Configuring the Target System As a Trusted Source"
- Section 4.5.6, "Reconciliation Rules for Oracle Database"
- Section 4.5.7, "Reconciliation Action Rules for Oracle Database"
- Section 4.5.8, "Performing Full Reconciliation from Oracle Database"
- Section 4.5.9, "Performing Limited Reconciliation from Oracle Database"
- Section 4.5.10, "Performing Batched Reconciliation from Oracle Database"
- Section 4.5.11, "Performing Incremental Reconciliation from Oracle Database"

#### 4.5.1 Guidelines on Configuring Reconciliation for Oracle Database

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.
4.5.2 Reconciliation Process for Oracle Database

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

See Also: The "Reconciliation" section in Oracle Fusion Middleware User's Guide for 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:
   a. Each user record fetched from the target system is compared with existing OIM Users. The reconciliation rule is applied during the comparison process. See Section 4.5.6, "Reconciliation Rules for Oracle Database" for information about the reconciliation rule.
   b. 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:
   a. 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. See Section 4.5.6, "Reconciliation Rules for Oracle Database" for information about the reconciliation rule.
   b. 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.

4.5.3 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. Section 4.7.1, "Guidelines on Configuring 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.
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 Section 4.3.3.2, "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 Section 4.3.3.3, "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. Depending on the Oracle Identity Manager release you are using, perform one of the following steps:
   - For Oracle Identity Manager release 11.1.1.x:
     Log in to the Administrative and User Console
   - For Oracle Identity Manager release 11.1.2.x or later:
     Log in to Oracle Identity System Administration

2. If you are using Oracle Identity Manager release 11.1.1.x, then:
   a. On the Welcome page, click **Advanced** in the upper-right corner of the page.
   b. On the Welcome to Oracle Identity Manager Advanced Administration page, in the Configuration region, click **Create IT Resource**.

3. If you are using Oracle Identity Manager release 11.1.2.x or later, then:
   a. In the left pane, under Configuration, click **IT Resource**.
   b. In the Manage IT Resource page, click **Create IT Resource**.

4. 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.

5. Click **Continue**.


  **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.

7. 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**: Oracle Fusion Middleware User’s Guide 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.

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.
4.5.7 Reconciliation Action Rules for Oracle Database

This section provides information on the reconciliation action rules for reconciliation.

No action is performed for rule conditions that are not predefined for this connector. You can define your own action rule for such rule conditions. See Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for information about modifying or creating reconciliation action rules.

Table 4–16 lists the action rules for target resource reconciliation.

<table>
<thead>
<tr>
<th>Rule Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Matches Found</td>
<td>Assign to Administrator With Least Load</td>
</tr>
<tr>
<td>One Entity Match Found</td>
<td>Establish Link</td>
</tr>
<tr>
<td>One Process Match Found</td>
<td>Establish Link</td>
</tr>
</tbody>
</table>

Table 4–17 lists the action rules for trusted source reconciliation.

<table>
<thead>
<tr>
<th>Rule Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Matches Found</td>
<td>Create User</td>
</tr>
<tr>
<td>One Entity Match Found</td>
<td>Establish Link</td>
</tr>
</tbody>
</table>

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.
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.

4.5.8 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 Section 4.4.2, "Attributes for Scheduled Jobs for Oracle Database" for information about this scheduled job.

### 4.5.9 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UID</strong></td>
<td>Unique identity representing the user. This parameter is mapped to USERNAME or <strong>NAME</strong> connector attribute.</td>
</tr>
<tr>
<td>authType</td>
<td>Authentication type of the user account. The value of this parameter can be one of the following: PASSWORD, GLOBAL, or EXTERNAL</td>
</tr>
<tr>
<td>tablespace</td>
<td>Default tablespace for user operations</td>
</tr>
<tr>
<td>defaultQuota</td>
<td>Quota for user operations on default tablespace. If no value is specified, the quota is set to unlimited.</td>
</tr>
<tr>
<td>globalDN</td>
<td>Unique name that identifies a user across an enterprise, if the authentication type is GLOBAL.</td>
</tr>
<tr>
<td><strong>ENABLE</strong></td>
<td>Status of the user account. The user is disabled if the value is one of following: LOCKED, EXPIRED, or LOCKED &amp; EXPIRED. The list of values for the disabled status is provided in the Lookup.DBUM.Oracle.Configuration lookup definition.</td>
</tr>
<tr>
<td>tempTableSpace</td>
<td>Temporary tablespace for user operations. Quota is always unlimited on temporary tablespace.</td>
</tr>
<tr>
<td>profile</td>
<td>Profile of the user account</td>
</tr>
<tr>
<td>lastModified</td>
<td>Last modified time-stamp. This parameter is used for incremental reconciliation operations.</td>
</tr>
</tbody>
</table>

For detailed information about ICF Filters, see the "ICF Filter Syntax" section of the *Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager*.
While deploying the connector, follow the instructions in Section 4.4.3, "Configuring Scheduled Jobs for Oracle Database" to specify attribute values.

4.5.10 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 Section 4.4.3, "Configuring Scheduled Jobs for Oracle Database."

4.5.11 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.

See Also: The "Provisioning" section in Oracle Fusion Middleware User’s Guide for Oracle Identity Manager for conceptual information about provisioning

This section contains the following topics about provisioning:

- Section 4.6.1, "Guidelines on Performing Provisioning Operations for Oracle Database"
- Section 4.6.2, "Provisioning Process for Oracle Database"
Section 4.6.3, "Configuring Direct Provisioning for Oracle Database"

Section 4.6.4, "Configuring Request-Based Provisioning for Oracle Database"

Section 4.6.5, "Switching Between Request-Based Provisioning and Direct Provisioning for Oracle Database"

Section 4.6.6, "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:

  ```sql
  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 a externally-authenticated database user:

  ```sql
  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:

  ```sql
  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:

  ```sql
  UNLIMITED
  ```

### 4.6.2 Provisioning Process for Oracle Database

**See Also:** The "Provisioning" section in Oracle Fusion Middleware User’s Guide for Oracle Identity Manager for conceptual information about provisioning

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.

This 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 Section 4.6.5, "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:
     \[ \text{GRANT JAVA_ADMIN TO jdoe WITH ADMIN OPTION} \]
   
   - If the administrator did not specify a value for granting role with the admin option, then the following SQL statement is formed:
     \[ \text{GRANT JAVA_ADMIN TO jdoe} \]

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.


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

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

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.

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

- Section 4.6.4.1, "Approver's Role in Request-Based Provisioning"
4.6.4.1 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 Importing Request Datasets Using Deployment Manager

See Also: Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager for detailed information about importing objects from an XML file using the 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.
4.6.4.3 End User's Role in Request-Based Provisioning

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

See Also: Oracle Fusion Middleware User’s Guide for Oracle Identity Manager for detailed information about these steps

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.

4.6.4.4 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.

### 4.6.4.5 Running the PurgeCache Utility

Run the PurgeCache utility to clear content belonging to the Metadata category from the server cache. See Section 2.3.1.3, "Clearing Content Related to 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

---

**Note:** It is assumed that you have performed the procedure described in Section 4.6.4, "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.

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

   Note: See the "Managing Users" chapter in Oracle Fusion Middleware User's Guide for 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 the "Managing Users" chapter in Oracle Fusion Middleware User's Guide for 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

The following sections describe procedures that you can perform to extend the functionality of the connector for addressing your specific business requirements:
4.7.1 Guidelines on Configuring 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.
The following sections discuss guidelines that you must apply while modifying the predefined queries or creating new queries:

- Section 4.7.1.1, "Syntax of Provisioning Queries for Oracle Database"
- Section 4.7.1.2, "Syntax of Reconciliation Queries for Oracle Database"
- Section 4.7.1.3, "Syntax of List of Values Queries for Oracle Database"
- Section 4.7.1.4, "Guidelines for Configuring Search Queries Used in Reconciliation from Oracle Database"

### 4.7.1.1 Syntax of Provisioning Queries for Oracle Database

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

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

For example:

```plaintext
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>
<thead>
<tr>
<th>Operation</th>
<th>Selection Logic</th>
<th>Query IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>CREATE_AUTHTYPE_OBJECTYPE</td>
<td>CREATE_PASSWORD_USER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CREATE_GLOBAL_USER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CREATE_EXTERNAL_USER</td>
</tr>
<tr>
<td>DELETE</td>
<td>DELETE_OBJECTTTYPE</td>
<td>DELETE_USER</td>
</tr>
<tr>
<td>ENABLE</td>
<td>ENABLE_OBJECTTTYPE</td>
<td>ENABLE_USER</td>
</tr>
<tr>
<td>DISABLE</td>
<td>DISABLE_OBJECTTTYPE</td>
<td>DISABLE_USER</td>
</tr>
<tr>
<td>RESET PASSWORD</td>
<td>SET_PASSWORD</td>
<td>SET_PASSWORD</td>
</tr>
</tbody>
</table>
4.7.1.2 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','ddmmyyyy')) *24*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
QUERY  PARAM1, PARAM2... [EXTENSIONJOIN [EXTENSION1
EXTENSIONSEPARATOR EXTENSION2 EXTENSIONSEPARATOR...]]
```

For example:

```sql
SELECT {__UID__}, {authType}, {externalname}, {tablespace}, {status},
{tempTableSpace}, {profile}, {defaultQuota}, {tmpQuota}, {lastModified}
FROM DBA_USERS dba {filter}, SEARCH_USER_ROLE, SEARCH_USER_PRIVILEGE
```

### 4.7.1.3 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:

```sql
OBJECTTYPE = "QUERY"
```

For example:

```sql
__PROFILE__="SELECT DISTINCT profile FROM dbaProfiles"
```

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.

### 4.7.1.4 Guidelines for Configuring Search Queries Used in Reconciliation from Oracle Database

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:

```java
"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:

- Section 4.7.2.1, "Updating the Query Files for Oracle Database"
- Section 4.7.2.2, "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:

   ```text
   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.

**See Also:** *Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager* for detailed information about the Download JARs utility

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 Section 4.7.1, "Guidelines on Configuring 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", "TEMPPTS_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:

   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.

   See Also: Oracle Fusion Middleware Developer’s Guide for Oracle
   Identity Manager for detailed information about the Update JARs
   utility

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. See Step 2 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for more information.

c. Create a new UI form to view the newly added field along with the rest of the fields. See Step 3 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for 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 4.c), and then save the application instance.

e. Publish the sandbox. See Step 5 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for more information.

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.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.

6. Create a process task to update the new field Temporary Quota as follows:


b. Double-click Process Definition and open the Oracle DB User process definition.

c. In the process definition, add a new task for updating the field as follows:
– Click **Add** and enter the task name, for example, *Temporary Quota Updated*, and the task description.

– In the Task Properties section, select the following fields:
  
  Conditional
  
  Allow Multiple Instances

– Click the save icon.

d. On the Integration tab, click **Add**, and then click **Adapter**.

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:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Map To</th>
<th>Qualifier</th>
<th>Literal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter return value</td>
<td>Object</td>
<td>Response code</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>attrName</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>Temporary Quota</td>
</tr>
<tr>
<td>ITResField</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_ORA_U_ITRES</td>
</tr>
<tr>
<td>newVal</td>
<td>String</td>
<td>Process Data</td>
<td>tmpQuota</td>
<td>NA</td>
</tr>
<tr>
<td>objectType</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>User</td>
</tr>
<tr>
<td>oldValue</td>
<td>String</td>
<td>Process Data</td>
<td>tmpQuota</td>
<td>NA</td>
</tr>
<tr>
<td><strong>Note:</strong> The old value check box must be selected.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Map To</th>
<th>Qualifier</th>
<th>Literal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>proInstance</td>
<td>Long</td>
<td>Process Data</td>
<td>Process Instance</td>
<td>NA</td>
</tr>
</tbody>
</table>

  
  g. On the Responses tab, click **Add** to add the following response codes:

<table>
<thead>
<tr>
<th>Code Name</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>Error occurred</td>
<td>R</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>An unknown response was received</td>
<td>R</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Operation completed</td>
<td>C</td>
</tr>
</tbody>
</table>

h. Click the save icon and then close the dialog box.

### 4.7.3 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.

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 Section 2.3.1.7, "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

<table>
<thead>
<tr>
<th>Connector Object</th>
<th>Name</th>
<th>Referenced By</th>
<th>Comments on Creating a Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT resource</td>
<td>Oracle</td>
<td>UD_DB_ORA_U (process form)</td>
<td>Create a copy of the IT resource with a different name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scheduled tasks</td>
<td></td>
</tr>
<tr>
<td>Resource object</td>
<td>Oracle DB User</td>
<td>All connector operations</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Oracle DB Trusted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scheduled Jobs</td>
<td>There are many scheduled jobs for different purposes.</td>
<td>NA</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Table 4–19 (Cont.) Connector Objects and Their Associations

<table>
<thead>
<tr>
<th>Connector Object Name</th>
<th>Referenced By</th>
<th>Comments on Creating a Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process definition</td>
<td>Oracle DB User</td>
<td>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. <strong>Note:</strong> Create copies of the process form only if there are differences in attributes between the various installations of the target system.</td>
</tr>
<tr>
<td>Process form</td>
<td>UD_DB_ORA_U</td>
<td>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. <strong>Note:</strong> Create copies of the process form only if there are differences in attributes between the various installations of the target system.</td>
</tr>
<tr>
<td>Child process form</td>
<td>UD_DB_ORA_R</td>
<td>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.</td>
</tr>
<tr>
<td>Configuration lookup definition for a target system configured as a target resource</td>
<td>Lookup.DBUM. Oracle.Configuration</td>
<td>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. <strong>Note:</strong> 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.</td>
</tr>
</tbody>
</table>
When you configure reconciliation:
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 perform provisioning operations:
When you use the Administrative and User Console 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 (Cont.) Connector Objects and Their Associations

<table>
<thead>
<tr>
<th>Connector Object</th>
<th>Name</th>
<th>Referenced By</th>
<th>Comments on Creating a Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration lookup definition for a target system configured as a trusted source</td>
<td>Lookup.DBUM. Oracle.Configura tion.Trusted</td>
<td>Oracle DB (IT resource)</td>
<td>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. <strong>Note:</strong> 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.</td>
</tr>
<tr>
<td>Resource object attributes mapping lookup definition (for target resource)</td>
<td>Lookup.DBUM. Oracle.UM.Recon AttrMap</td>
<td>NA</td>
<td>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. <strong>Note:</strong> Create copies of this lookup definition only if there are differences in attributes between the two installations of the target system.</td>
</tr>
<tr>
<td>Configuration lookup definition for a target system configured as a trusted source</td>
<td>Lookup.DBUM. Oracle.UM.Recon AttrMap.Trusted</td>
<td>Oracle DB (IT resource)</td>
<td>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. <strong>Note:</strong> 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.</td>
</tr>
</tbody>
</table>
4.7.4 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 Oracle Fusion Middleware Developer's Guide for 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 (#):

   ```java
   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
   ```
Extending the Connector for Oracle Database

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 Section 4.3.6, "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.

See Also: Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for detailed information about the Upload JARs utility

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 {
```

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.
Extending the Connector for Oracle Database

/*
* 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 Section 4.3.5, "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.

**See Also:** Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for detailed information about the Upload JARs utility

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 Section 4.3.4, "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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>User001</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>User001</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserName[Pattern]</td>
<td>User001</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserName[Pattern]</td>
<td>00012*</td>
</tr>
</tbody>
</table>

6. Click the save icon.

### 4.7.8 Configuring 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.
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.

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:

   ```sql
   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

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.

- Section 5.1, "Configuring Secure Communication Between MySQL and Oracle Identity Manager"
- Section 5.2, "Determining Values for the JDBC URL and Connection Properties Parameters for MySQL"
- Section 5.3, "Lookup Definitions for MySQL"
- Section 5.4, "Scheduled Jobs for MySQL"
- Section 5.5, "Reconciliation from MySQL"
- Section 5.6, "Provisioning for MySQL"
- Section 5.7, "Extending the Connector for MySQL"

### 5.1 Configuring Secure Communication Between MySQL and Oracle Identity Manager

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

<table>
<thead>
<tr>
<th>Application Server</th>
<th>Truststore Location</th>
</tr>
</thead>
</table>
| Oracle WebLogic Server   | - If you are using Oracle jrockit_R27.3.1-jdk, then import the certificate into the keystore in the following directory: JROCKIT_HOME/jre/lib/security  
                        | - If you are using the default Oracle WebLogic Server JDK, then import the certificate into the keystore in following directory: WEBLOGIC_HOME/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: JAVA_HOME/jre/lib/security/cacerts |

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 Section 2.3.4, “Configuring the IT Resource for the Connector Server.”

5.2  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 Section 2.3.2, “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

Lookup definitions used during connector operations can be categorized as follows:

- **Section 5.3.1, "Lookup Definitions Synchronized with MySQL"**
You must provide Decode values for some of the entries of the following lookup definitions. 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.

### 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 on the Administrative and User Console, 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>
<thead>
<tr>
<th>Lookup Definition</th>
<th>Target Column Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup.DBUM.MySQL.SchemaPrivileges</td>
<td>Privilege</td>
</tr>
</tbody>
</table>

### 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:

- Section 5.3.2.1, "Lookup.DBUM.MySQL.Configuration"
- Section 5.3.2.2, "Lookup.DBUM.MySQL.UM.Configuration"
- Section 5.3.2.3, "Lookup.DBUM.MySQL.Configuration.Trusted"
- Section 5.3.2.4, "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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bundle Name</td>
<td>org.identityconnectors.dbum</td>
<td>Name of the connector bundle package. Do not modify this entry.</td>
</tr>
<tr>
<td>Bundle Version</td>
<td>1.0.1116</td>
<td>Version of the connector bundle class. Do not modify this entry.</td>
</tr>
<tr>
<td>Connector Name</td>
<td>org.identityconnectors.dbum.DBUMConnector</td>
<td>Name of the connector class. Do not modify this entry.</td>
</tr>
<tr>
<td>User Configuration Lookup</td>
<td>Lookup.DBUM.MySQL.UM.Configuration</td>
<td>Name of the lookup definition that contains user-specific configuration properties. Do not modify this entry.</td>
</tr>
</tbody>
</table>

#### 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.
### 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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bundle Name</td>
<td>org.identityconnectors.dbum</td>
</tr>
<tr>
<td>Bundle Version</td>
<td>1.0.1116</td>
</tr>
<tr>
<td>Connector Name</td>
<td>org.identityconnectors.dbum.DBUMConnector</td>
</tr>
<tr>
<td>User Configuration Lookup</td>
<td>Lookup.DBUM.MySQL.UM.Configuration.Trusted</td>
</tr>
</tbody>
</table>

### 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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recon Attribute Defaults</td>
<td>Lookup.DBUM.MySQL.UM.ReconDefaults.Trusted</td>
</tr>
<tr>
<td>Recon Attribute Map</td>
<td>Lookup.DBUM.MySQL.UM.ReconAttrMap.Trusted</td>
</tr>
<tr>
<td>Recon Validation Lookup</td>
<td>Lookup.DBUM.MySQL.UM.ReconValidations.Trusted</td>
</tr>
<tr>
<td>Recon Exclusion List</td>
<td>Lookup.DBUM.MySQL.UM.ReconExclusionList.Trusted</td>
</tr>
<tr>
<td>Recon Transformation Lookup</td>
<td>Lookup.DBUM.MySQL.UM.ReconTransformations.Trusted</td>
</tr>
</tbody>
</table>

### 5.3.3 Lookup Definitions for Attribute Mappings for MySQL

This section describes the following lookup definitions:

- Section 5.3.3.1, "Lookup.DBUM.MySQL.UM.ProvAttrMap"
- Section 5.3.3.2, "Lookup.DBUM.MySQL.UM.ReconAttrMap"
- Section 5.3.3.3, "Lookup.DBUM.MySQL.UM.ReconAttrMap.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>
<thead>
<tr>
<th>Table 5–7 Entries in Lookup.DBUM.MySQL.UM.ProvAttrMap</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Code Key</strong></td>
</tr>
<tr>
<td>Return Id</td>
</tr>
<tr>
<td>UD_DB_MYS P~PrivilegeLOOKUP</td>
</tr>
<tr>
<td>User Name</td>
</tr>
<tr>
<td>User Password</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Table 5–8 Entries in Lookup.DBUM.MySQL.UM.ReconAttrMap</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Code Key</strong></td>
</tr>
<tr>
<td>Privilege List~PrivilegeNameLOOKUP</td>
</tr>
<tr>
<td>Return ID</td>
</tr>
<tr>
<td>User Name</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Table 5–9 Entries in Lookup.DBUM.MySQL.UM.ReconAttrMap.Trusted</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Code Key</strong></td>
</tr>
<tr>
<td>First Name</td>
</tr>
<tr>
<td>User ID</td>
</tr>
</tbody>
</table>

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)
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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>root</td>
</tr>
</tbody>
</table>

- For trusted source mode: Lookup.DBUM.MySQL.UM.ExclusionList.Trusted
  By default, this lookup definition has the following entry:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID</td>
<td>root</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
<th>Sample Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>User ID of a user</td>
<td>Code Key: User Name \n Decode: User001</td>
</tr>
<tr>
<td>User Name with the [PATTERN] suffix</td>
<td>A regular expression supported by the representation in the java.util.regex.Pattern class</td>
<td>Code Key: User Name[PATTERN] \n To exclude users matching any of the user ID’s User001, User002, User088, then: \n Decode: User001</td>
</tr>
</tbody>
</table>

See Also: For information about the supported patterns, visit [http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html](http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html)
Section 5.7.7, "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

Section 5.7.6, "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.

Section 5.7.5, "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:

■ Section 5.4.1, "Scheduled Job for Lookup Field Synchronization for MySQL"
■ Section 5.4.2, "Attributes for Scheduled Jobs for MySQL"
■ Section 5.4.3, "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 following scheduled job is used for lookup field synchronization:

■ DBUM MySQL Privilege Type Lookup Reconciliation

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Key Attribute</td>
<td>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.</td>
</tr>
<tr>
<td>Decode Attribute</td>
<td>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></td>
</tr>
<tr>
<td>IT Resource Name</td>
<td>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></td>
</tr>
<tr>
<td>Lookup Name</td>
<td>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></td>
</tr>
<tr>
<td>Object Type</td>
<td>Enter the type of object whose values must be synchronized. Default value: <code>__PRIVILEGES__</code>  Note: Do not change the value of this attribute.</td>
</tr>
<tr>
<td>Resource Object Name</td>
<td>Enter the name of the resource object that is used for reconciliation. Default value: <code>MySQL DB User</code></td>
</tr>
</tbody>
</table>

### 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.
### 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 Section 5.4.1, "Scheduled Job for Lookup Field Synchronization for MySQL" and Section 5.4.2, "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. Depending on the Oracle Identity Manager release you are using, perform one of the following steps:
   - 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**.
   - For 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**.

2. Search for and open the scheduled job as follows:
   - 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.
   - In the search results table on the left pane, click the scheduled job in the Job Name column.

3. 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 *Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager* for detailed information about schedule types.

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

4. 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 Section 5.4.2, "Attributes for Scheduled Jobs for MySQL."

5. After specifying the attributes, click **Apply** to save the changes.
5.5 Reconciliation from MySQL

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:

- Section 5.5.1, "Guidelines on Configuring Reconciliation for MySQL"
- Section 5.5.2, "Reconciliation Process for MySQL"
- Section 5.5.3, "Target System Columns Used in Reconciliation from MySQL"
- Section 5.5.4, "Configuring the Target System As a Trusted Source"
- Section 5.5.5, "Reconciliation Rules for MySQL"
- Section 5.5.6, "Reconciliation Action Rules for MySQL"
- Section 5.5.7, "Performing Full Reconciliation from MySQL"
- Section 5.5.8, "Performing Limited Reconciliation from MySQL"
- Section 5.5.9, "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 Reconciliation Process for MySQL

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

See Also: The "Reconciliation" section in Oracle Fusion Middleware User’s Guide for 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.

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:
   
   a. Each user record fetched from the target system is compared with existing OIM Users. The reconciliation rule is applied during the comparison process. See Section 5.5.5, "Reconciliation Rules for MySQL" for information about the reconciliation rule.

   b. 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:
   
   a. 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. See Section 5.5.5, "Reconciliation Rules for MySQL" for information about the reconciliation rule.

   b. 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.

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 Section 5.3.3.2, "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 Section 5.3.3.3, "Lookup.DBUM.MySQL.UM.ReconAttrMap.Trusted" for more information about this lookup definition.

5.5.4 Configuring the Target System As a Trusted Source

To configure trusted source reconciliation:

1. Depending on the Oracle Identity Manager release you are using, perform one of the following steps:
   - For Oracle Identity Manager release 11.1.1.x:
     Log in to the Administrative and User Console
   - For Oracle Identity Manager release 11.1.2.x or later:
     Log in to Oracle Identity System Administration

2. If you are using Oracle Identity Manager release 11.1.1.x, then:
   a. On the Welcome page, click Advanced in the upper-right corner of the page.
   b. On the Welcome to Oracle Identity Manager Advanced Administration page, in the Configuration region, click Create IT Resource.

3. If you are using Oracle Identity Manager release 11.1.2.x or later, then:
   a. In the left pane, under Configuration, click IT Resource.
   b. In the Manage IT Resource page, click Create IT Resource.

4. 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.

5. Click Continue.

   - 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.

7. 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: Oracle Fusion Middleware User’s Guide 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.

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.
4. Search for the rule name.

### 5.5.6 Reconciliation Action Rules for MySQL

This section provides information on the reconciliation action rules for reconciliation.

No action is performed for rule conditions that are not predefined for this connector. You can define your own action rule for such rule conditions. See Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for information about modifying or creating reconciliation action rules.

**Table 5–14** lists the action rules for target resource reconciliation.
Table 5–14  Action Rules for Target Resource Reconciliation

<table>
<thead>
<tr>
<th>Rule Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Matches Found</td>
<td>Assign to Administrator With Least Load</td>
</tr>
<tr>
<td>One Entity Match Found</td>
<td>Establish Link</td>
</tr>
<tr>
<td>One Process Match Found</td>
<td>Establish Link</td>
</tr>
</tbody>
</table>

Table 5–15 lists the action rules for trusted source reconciliation.

Table 5–15  Action Rules for Trusted Source Reconciliation

<table>
<thead>
<tr>
<th>Rule Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Matches Found</td>
<td>Create User</td>
</tr>
<tr>
<td>One Entity Match Found</td>
<td>Establish Link</td>
</tr>
</tbody>
</table>

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.
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.

5.5.7 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 Section 5.4.2, "Attributes for Scheduled Jobs for MySQL" for information about this scheduled job.

5.5.8 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UID</strong></td>
<td>Unique identity representing the user</td>
</tr>
<tr>
<td></td>
<td>This parameter is mapped to USERNAME or <strong>NAME</strong> connector attribute.</td>
</tr>
</tbody>
</table>

For detailed information about ICF Filters, see the "ICF Filter Syntax" section of the Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager.

While deploying the connector, follow the instructions in Section 5.4.3, "Configuring Scheduled Jobs for MySQL" to specify attribute values.

### 5.5.9 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 Section 5.4.3, "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.
See Also: The "Provisioning" section in Oracle Fusion Middleware User's Guide for Oracle Identity Manager for conceptual information about provisioning

This section contains the following topics about provisioning:

- Section 5.6.1, "Guidelines on Performing Provisioning Operations for MySQL"
- Section 5.6.2, "Provisioning Process for MySQL"
- Section 5.6.3, "Configuring Direct Provisioning for MySQL"
- Section 5.6.4, "Configuring Request-Based Provisioning for MySQL"
- Section 5.6.5, "Switching Between Request-Based Provisioning and Direct Provisioning for MySQL"
- Section 5.6.6, "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 Provisioning Process for MySQL

See Also: The "Provisioning" section in Oracle Fusion Middleware User’s Guide for Oracle Identity Manager for conceptual information about provisioning

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.

This 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 Section 5.6.5, "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: 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 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.
   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 MySQL DB User resource from the list, and then 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.

5.6.4 Configuring Request-Based Provisioning for MySQL

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

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.

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

- Section 5.6.4.1, "Approver’s Role in Request-Based Provisioning"
- Section 5.6.4.2, "Importing Request Datasets Using Deployment Manager"
- Section 5.6.4.3, "End User’s Role in Request-Based Provisioning"
- Section 5.6.4.4, "Enabling the Auto Save Form Feature"
- Section 5.6.4.5, "Running the PurgeCache Utility"

5.6.4.1 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.
5.6.4.2 Importing Request Datasets Using Deployment Manager

See Also: Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager for detailed information about importing objects from an XML file using the 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.

5.6.4.3 End User’s Role in Request-Based Provisioning

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

See Also: Oracle Fusion Middleware User’s Guide for Oracle Identity Manager for detailed information about these steps

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.

5.6.4.4 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 MySQL DB process definition.

4. Select the Auto Save Form check box.

5. Click the save icon.

5.6.4.5 Running the PurgeCache Utility
Run the PurgeCache utility to clear content belonging to the Metadata category from the server cache. See Section 2.3.1.3, "Clearing Content Related to 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

Note: It is assumed that you have performed the procedure described in Section 5.6.4, "Configuring Request-Based Provisioning for MySQL."
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 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.

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

    Note: See the "Managing Users" chapter in Oracle Fusion Middleware User’s Guide for 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 the "Managing Users" chapter in Oracle Fusion Middleware User’s Guide for 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 the “Managing Lookups” chapter of the Oracle Fusion Middleware Administering Oracle Identity Manager guide for information about managing lookups by using the Form Designer in the Oracle Identity Manager System Administration console.

- Section 5.7.1, "Guidelines on Configuring the Queries for MySQL"
- Section 5.7.2, "Configuring Queries to Add Support for Custom Parameters and Lookup Fields for MySQL"
- Section 5.7.3, "Configuring the Connector for Multiple Installations of MySQL"
- Section 5.7.4, "Configuring the Connector for Multiple Trusted Source Reconciliation from MySQL"
- Section 5.7.5, "Configuring Validation of Data During Reconciliation and Provisioning for MySQL"
5.7.1 Guidelines on Configuring the Queries for MySQL

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 table spaces. 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.

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

- **Section 5.7.1.1, "Syntax of Provisioning Queries for MySQL Database"**
- **Section 5.7.1.2, "Syntax of Reconciliation Queries for MySQL Database"**
- **Section 5.7.1.3, "Syntax of List of Values Queries for MySQL Database"**

5.7.1.1 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 **EXTENSIONSEP ARATOR**.

  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>
<thead>
<tr>
<th>Operation</th>
<th>Selection Logic</th>
<th>Query IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>CREATE_OBJECTTYPE</td>
<td>CREATE_USER</td>
</tr>
<tr>
<td>DELETE</td>
<td>DELETE_OBJECTTYPE</td>
<td>DELETE_USER</td>
</tr>
<tr>
<td>RESET PASSWORD</td>
<td>SET_PASSWORD</td>
<td>SET_PASSWORD</td>
</tr>
<tr>
<td>ADD CHILD VALUES</td>
<td>UPDATE_ADD_ATTRIBUTE</td>
<td>UPDATE_ADD_PRIVILEGES</td>
</tr>
<tr>
<td>REMOVE CHILD VALUES</td>
<td>UPDATE_REVOKE_ATTRIBUTE</td>
<td>UPDATE_REVOKE_PRIVILEGES</td>
</tr>
</tbody>
</table>

5.7.1.2 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
Extending the Connector for MySQL

- **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
  ```

5.7.1.3 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"
```
Extending the Connector for MySQL

For example:

```java
__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:

- **Section 5.7.2.1, "Updating the Query Files for MySQL Database"**
- **Section 5.7.2.2, "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:

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

   For Microsoft Windows:
   ```bash
   OIM_HOME/server/bin/DownloadJars.bat
   ```

   For UNIX:
   ```bash
   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.

See Also: Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager for detailed information about the Download JARs utility

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 Section 5.7.1, "Guidelines on Configuring the 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:
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:

```
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.

For Microsoft Windows:

```
OIM_HOME/server/bin/UpdateJars.bat
```

For UNIX:

```
OIM_HOME/server/bin/UpdateJars.sh
```

See Also: Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for detailed information about the Update JARs utility
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. See Step 2 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for more information.
   c. Create a new UI form to view the newly added field along with the rest of the fields. See Step 3 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for 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 4.c), and then save the application instance.
   e. Publish the sandbox. See Step 5 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for more information.

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:
   b. Double-click Process Definition and open the MySQL DB User process definition.
   c. In the process definition, add a new task for updating the field as follows:
      - Click Add and enter the task name, for example, Custom Attribute 1 Updated, and the task description.
– In the Task Properties section, select the following fields:

  Conditional
  Allow Multiple Instances
– Click the save icon.

d. On the Integration tab, click Add, and then click Adapter.

e. Select the `adpMYSQLDBUMUPDATEUSER` 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:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Map To</th>
<th>Qualifier</th>
<th>Literal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter return value</td>
<td>Object</td>
<td>Response code</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>attributeName</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>Custom Attribute 1</td>
</tr>
<tr>
<td>itRes</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_MYS_U_ITRES</td>
</tr>
<tr>
<td>objectType</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>User</td>
</tr>
<tr>
<td>processInstanceKey</td>
<td>Long</td>
<td>Process Data</td>
<td>Process Instance</td>
<td>NA</td>
</tr>
</tbody>
</table>


- On the Responses tab, click Add to add the following response codes:

<table>
<thead>
<tr>
<th>Code Name</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>Error occurred</td>
<td>R</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>An unknown response was received</td>
<td>R</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Operation completed</td>
<td>C</td>
</tr>
</tbody>
</table>

h. Click the save icon and then close the dialog box.

5.7.3 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.

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 Section 2.3.1.7, "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

<table>
<thead>
<tr>
<th>Connector Object</th>
<th>Name</th>
<th>Referenced By</th>
<th>Comments on Creating a Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT resource</td>
<td>MySQL DB</td>
<td>UD_DB_MYS_U (process form)</td>
<td>Create a copy of the IT resource with a different name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scheduled tasks</td>
<td></td>
</tr>
<tr>
<td>Resource object</td>
<td>MySQL DB User</td>
<td>All connector operations</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>MySQL DB Trusted</td>
<td></td>
<td>Note: Create copies of the resource object only if there are differences in attributes between the various installations of the target system.</td>
</tr>
<tr>
<td>Scheduled Jobs</td>
<td>There are many scheduled jobs for different purposes.</td>
<td>NA</td>
<td>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.</td>
</tr>
<tr>
<td>Process definition</td>
<td>MySQL DB User</td>
<td>NA</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Note: Create copies of the process form only if there are differences in attributes between the various installations of the target system.</td>
</tr>
</tbody>
</table>
### Table 5–17  (Cont.) Connector Objects and Their Associations

<table>
<thead>
<tr>
<th>Connector Object</th>
<th>Name</th>
<th>Referenced By</th>
<th>Comments on Creating a Copy</th>
</tr>
</thead>
</table>
| Process form                                          | UD_DB_MYS_U           | MySQL DB User (Process definition)   | 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.  
Note: Create copies of the process form only if there are differences in attributes between the various installations of the target system. |
| Child process form                                    | UD_DB_MYS_P           | MySQL DB User (Process definition)   | 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. |
|                                                       | UD_DB_MYS_U           | MySQL DB User (Process form)         |                              |
| Configuration lookup definition for a target system configured as a target resource | Lookup.DBUM. MySQL.Configuration | MySQL 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. |
When you configure reconciliation:
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 perform provisioning operations:**
When you use the Administrative and User Console 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 (Cont.) Connector Objects and Their Associations

<table>
<thead>
<tr>
<th>Connector Object Name</th>
<th>Referenced By</th>
<th>Comments on Creating a Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration lookup definition for a target system configured as a trusted source</td>
<td>Lookup.DBUM. MySQL.Config. Trusted MySQL DB (IT resource)</td>
<td>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. <strong>Note:</strong> 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.</td>
</tr>
<tr>
<td>Resource object attributes mapping lookup definition (for target resource)</td>
<td>Lookup.DBUM. MySQL.UM.ReconAttrMap NA</td>
<td>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. <strong>Note:</strong> Create copies of this lookup definition only if there are differences in attributes between the two installations of the target system.</td>
</tr>
<tr>
<td>Configuration lookup definition for a target system configured as a trusted source</td>
<td>Lookup.DBUM. MySQL.UM. ReconAttrMap.Trusted MySQL DB (IT resource)</td>
<td>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. <strong>Note:</strong> 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.</td>
</tr>
</tbody>
</table>
5.7.4 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 Oracle Fusion Middleware Developer’s Guide for 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 (#):

```java
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 Section 5.3.6, "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.

**See Also:** Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for detailed information about the Upload JARs utility

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 Section 5.3.5, "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.DBUMTransfomation.
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.

See Also: Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for detailed information about the Upload JARs utility

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 Section 5.3.4, "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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>User001</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>User001</td>
</tr>
</tbody>
</table>

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](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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name[PATTERN]</td>
<td>User001</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name[PATTERN]</td>
<td>00012*</td>
</tr>
</tbody>
</table>

6. Click the save icon.

### 5.7.8 Configuring 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.

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

This chapter contains the following topics:

- Section 6.1, "Configuring Secure Communication Between DB2 and Oracle Identity Manager"
- Section 6.2, "Determining Values for the JDBC URL and Connection Properties Parameters for DB2"
- Section 6.3, "Lookup Definitions for DB2"
- Section 6.4, "Scheduled Jobs for DB2"
- Section 6.5, "Reconciliation from DB2"
- Section 6.6, "Provisioning for DB2"
- Section 6.7, "Extending the Connector for DB2"

6.1 Configuring Secure Communication Between DB2 and Oracle Identity Manager

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:
Determining Values for the JDBC URL and Connection Properties Parameters for DB2

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 Section 2.3.4, "Configuring the IT Resource for the Connector Server."

6.2 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 Section 2.3.2, "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.

---

PRIORITY_TRUE

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

<table>
<thead>
<tr>
<th>Application Server</th>
<th>Truststore Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle WebLogic Server</td>
<td>- If you are using Oracle jrockit_R27.3.1-jdk, then import the certificate into the</td>
</tr>
<tr>
<td></td>
<td>keystore in the following directory:</td>
</tr>
<tr>
<td></td>
<td>JROCKIT_HOME/jre/lib/security</td>
</tr>
<tr>
<td></td>
<td>- If you are using the default Oracle WebLogic Server JDK, then import the</td>
</tr>
<tr>
<td></td>
<td>certificate into the keystore in following directory:</td>
</tr>
<tr>
<td></td>
<td>WEBLOGIC_HOME/java/jre/lib/security/cacerts</td>
</tr>
<tr>
<td></td>
<td>- If you are using a JDK other than Oracle jrockit_R27.3.1-jdk or Oracle</td>
</tr>
<tr>
<td></td>
<td>WebLogic Server JDK, then import the certificate into your keystore at the</td>
</tr>
<tr>
<td></td>
<td>following directory:</td>
</tr>
<tr>
<td></td>
<td>JAVA_HOME/jre/lib/security/cacerts</td>
</tr>
</tbody>
</table>

---
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:

\[\{\text{PROPERTY}=\text{VALUE}\} \ldots\] . . .

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

Lookup definitions used during connector operations can be categorized as follows:

- Section 6.3.1, "Lookup Definitions Synchronized with DB2"
- Section 6.3.2, "Lookup Definitions for Configurations for DB2"
- Section 6.3.3, "Lookup Definitions for Attribute Mappings for DB2"
- Section 6.3.4, "Lookup Definitions for Exclusion Lists for DB2"
- Section 6.3.5, "Lookup Definitions for Transformation of Data in DB2"
- Section 6.3.6, "Lookup Definition for Validation of Data in DB2"

You must provide Decode values for some of the entries of the following lookup definitions. 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.

### 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:** `ITRESOURCE_NAME~LOOKUP_FIELD_ID`
  
  In this format:
  - `ITRESOURCE_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 on the Administrative and User Console, 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>
<thead>
<tr>
<th>Lookup Definition</th>
<th>Target Column Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup.DBUM.DB2.Schema</td>
<td>Schema</td>
</tr>
<tr>
<td>Lookup.DBUM.DB2.Tablespaces</td>
<td>Tablespace</td>
</tr>
<tr>
<td>Lookup.DBUM.DB2.UserType</td>
<td>User Type</td>
</tr>
</tbody>
</table>

By default, the `Lookup.DBUM.DB2.UserType` lookup definition contains the following entries for user types:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP</td>
<td>GROUP</td>
</tr>
<tr>
<td>USER</td>
<td>USER</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>WITH GRANT OPTION</td>
<td>WITH GRANT OPTION</td>
</tr>
</tbody>
</table>

### 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:

- Section 6.3.2.1, "Lookup.DBUM.DB2.Configuration"
- Section 6.3.2.2, "Lookup.DBUM.DB2.UM.Configuration"
- Section 6.3.2.3, "Lookup.DBUM.DB2.Configuration.Trusted"
- Section 6.3.2.4, "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>
<thead>
<tr>
<th>Table 6–3 Entries in Lookup.DBUM.DB2.Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Code Key</strong></td>
</tr>
</tbody>
</table>
| 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

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provisioning Attribute Map</td>
<td>Lookup.DBUM.DB2.UM.ProvAttrMap</td>
</tr>
<tr>
<td>Provisioning Exclusion List</td>
<td>Lookup.DBUM.DB2.UM.ExclusionList</td>
</tr>
<tr>
<td>Provisioning Validation Lookup</td>
<td>Lookup.DBUM.DB2.UM.ProvValidations</td>
</tr>
<tr>
<td>Recon Validation Lookup</td>
<td>Lookup.DBUM.DB2.UM.ReconValidations</td>
</tr>
<tr>
<td>Recon Attribute Map</td>
<td>Lookup.DBUM.DB2.UM.ReconAttrMap</td>
</tr>
<tr>
<td>Recon Exclusion List</td>
<td>Lookup.DBUM.DB2.UM.ExclusionList</td>
</tr>
<tr>
<td>Recon Transformation Lookup</td>
<td>Lookup.DBUM.DB2.UM.ReconTransformations</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bundle Name</td>
<td>org.identityconnectors.dbum</td>
</tr>
<tr>
<td>Bundle Version</td>
<td>1.0.1116</td>
</tr>
<tr>
<td>Connector Name</td>
<td>org.identityconnectors.dbum.DBUMConnector</td>
</tr>
<tr>
<td>disableValuesSet</td>
<td>&quot;LOCKED&quot;</td>
</tr>
<tr>
<td>User Configuration Lookup</td>
<td>Lookup.DBUM.DB2.UM.Configuration.Trusted</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recon Attribute Defaults</td>
<td>Lookup.DBUM.DB2.UM.ReconDefaults.Trusted</td>
</tr>
<tr>
<td>Recon Attribute Map</td>
<td>Lookup.DBUM.DB2.UM.ReconAttrMap.Trusted</td>
</tr>
<tr>
<td>Recon Exclusion List</td>
<td>Lookup.DBUM.DB2.UM.ExclusionList.Trusted</td>
</tr>
<tr>
<td>Recon Transformation Lookup</td>
<td>Lookup.DBUM.DB2.UM.ReconTransformations.Trusted</td>
</tr>
</tbody>
</table>

6.3.3 Lookup Definitions for Attribute Mappings for DB2
This section describes the following lookup definitions:
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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return Id</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>UD_DB_DB2_S~Schema[LOOKUP]</td>
<td>schemas<del>DBSchema</del><strong>NAME</strong></td>
</tr>
<tr>
<td>UD_DB_DB2_S~Schema</td>
<td>schemas<del>DBSchema</del>grantOption</td>
</tr>
<tr>
<td>UD_DB_DB2_T~Tablespace[LOOKUP]</td>
<td>tables<del>DBTables</del><strong>NAME</strong></td>
</tr>
<tr>
<td>UD_DB_DB2_T~Tablespace</td>
<td>tables<del>DBTables</del>grantOption</td>
</tr>
<tr>
<td>Username</td>
<td><strong>NAME</strong></td>
</tr>
<tr>
<td>User Type</td>
<td>userType</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return ID</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>Schema List~Schema Grant option</td>
<td>schemas<del>DBSchema</del>grantOption</td>
</tr>
<tr>
<td>Schema List~Schema Name[LOOKUP]</td>
<td>schemas<del>DBSchema</del><strong>NAME</strong></td>
</tr>
<tr>
<td>Status</td>
<td><strong>ENABLE</strong></td>
</tr>
<tr>
<td>Tablespace List~Tablespace Grant Option</td>
<td>tables<del>DBTables</del>grantOption</td>
</tr>
<tr>
<td>Tablespace List~Tablespace Name[LOOKUP]</td>
<td>tables<del>DBTables</del><strong>NAME</strong></td>
</tr>
<tr>
<td>UserName</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>User Type</td>
<td>userType</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>User ID</td>
<td><strong>UID</strong></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empl Type</td>
<td>Full-Time</td>
</tr>
<tr>
<td>Organization Name</td>
<td>Xellerate Users</td>
</tr>
<tr>
<td>Status</td>
<td>Active</td>
</tr>
<tr>
<td>User Type</td>
<td>End-User</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>db2admin</td>
</tr>
</tbody>
</table>

- For trusted source mode: Lookup.DBUM.DB2.UM.ExclusionList.Trusted

  By default, this lookup definition has the following entry:
The following is the format of the values stored in these lookups:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
<th>Sample Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID</td>
<td><code>db2admin</code></td>
<td></td>
</tr>
</tbody>
</table>

Section 6.7.7, "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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema List</td>
<td><code>db2.iam.connectors.dbum.transformations.SchemaGrantTransformation</code></td>
</tr>
<tr>
<td>Tablespace List</td>
<td><code>db2.iam.connectors.dbum.transformations.TablespaceGrantTransformation</code></td>
</tr>
<tr>
<td>User Type</td>
<td><code>db2.iam.connectors.dbum.transformations.UserTypeTransformation</code></td>
</tr>
</tbody>
</table>

- **For trusted source mode:** `Lookup.DBUM.DB2.UM.ReconTransformations.Trusted`

  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](http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html)

Section 6.7.6, "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.
Section 6.7.5, "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:
- Section 6.4.1, "Scheduled Jobs for Lookup Field Synchronization for DB2"
- Section 6.4.2, "Attributes for Scheduled Jobs for DB2"
- Section 6.4.3, "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 Tables Spaces 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>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Key Attribute</td>
<td>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: <strong>NAME</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Do not change the value of this attribute.</td>
</tr>
<tr>
<td>Decode Attribute</td>
<td>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: <strong>NAME</strong></td>
</tr>
</tbody>
</table>
| 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: DB2
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>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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:  
- For DBUM DB2 Schema Lookup Reconciliation - Lookup.DBUM.DB2.Schema  
- For DBUM DB2 Tablespaces Lookup Reconciliation - Lookup.DBUM.DB2.Tablespaces |
| 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:  
- For DBUM DB2 Schema Lookup Reconciliation - __SCHEMAS__  
- For DBUM DB2 Tablespaces Lookup Reconciliation - __TABLESPACES__  
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: DB2 DB User |
| 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: equalTo('__UID__','SEPT12USER1') See Section 6.5.8, "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. |
You can apply this procedure to configure the scheduled jobs for lookup fields synchronization and reconciliation.

See Section 6.4.1, “Scheduled Jobs for Lookup Field Synchronization for DB2” and Section 6.4.2, “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. Depending on the Oracle Identity Manager release you are using, perform one of the following steps:
   - 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.
   - For 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.

2. Search for and open the scheduled job as follows:
   - 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.
   - In the search results table on the left pane, click the scheduled job in the Job Name column.

3. 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.

### Table 6–12 (Cont.) Attributes of the Scheduled Jobs for Reconciliation

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Type</td>
<td>Type of object you want to reconcile</td>
</tr>
<tr>
<td>Default value: User</td>
<td>Name of the resource object that is used for reconciliation</td>
</tr>
<tr>
<td>Resource Object Name</td>
<td>For DBUM DB2 User Target Reconciliation: DB2 DB User</td>
</tr>
<tr>
<td></td>
<td>For DBUM DB2 User Trusted Reconciliation: DB2 DB Trusted</td>
</tr>
<tr>
<td>Scheduled Task Name</td>
<td>Name of the scheduled job</td>
</tr>
</tbody>
</table>
| 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.
In addition to modifying the job details, you can enable or disable a job.

4. 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 Section 6.4.2, "Attributes for Scheduled Jobs for DB2."

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

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

### 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:

- Section 6.5.1, "Guidelines on Configuring Reconciliation for DB2"
- Section 6.5.2, "Reconciliation Process for DB2"
- Section 6.5.3, "Target System Columns Used in Reconciliation from DB2"
- Section 6.5.4, "Configuring the Target System As a Trusted Source"
- Section 6.5.5, "Reconciliation Rules for DB2"
- Section 6.5.6, "Reconciliation Action Rules for DB2"
- Section 6.5.7, "Performing Full Reconciliation from DB2"
- Section 6.5.8, "Performing Limited Reconciliation from DB2"
- Section 6.5.9, "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.
6.5.2 Reconciliation Process for DB2

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

See Also: The "Reconciliation" section in Oracle Fusion Middleware User's Guide for 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.

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:
   a. Each user record fetched from the target system is compared with existing OIM Users. The reconciliation rule is applied during the comparison process. See Section 6.5.5, "Reconciliation Rules for DB2" for information about the reconciliation rule.
   b. 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:
   a. 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. See Section 6.5.5, "Reconciliation Rules for DB2" for information about the reconciliation rule.
   b. 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.

### 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 Section 6.3.3.2, "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 Section 6.3.3.3, "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. Depending on the Oracle Identity Manager release you are using, perform one of the following steps:
   - For Oracle Identity Manager release 11.1.1.x:
     
     Log in to the Administrative and User Console
   - For Oracle Identity Manager release 11.1.2.x or later:
     
     Log in to Oracle Identity System Administration

2. If you are using Oracle Identity Manager release 11.1.1.x, then:
   
   a. On the Welcome page, click **Advanced** in the upper-right corner of the page.
   
   b. On the Welcome to Oracle Identity Manager Advanced Administration page, in the Configuration region, click **Create IT Resource**.
3. If you are using Oracle Identity Manager release 11.1.2.x or later, then:
   a. In the left pane, under Configuration, click **IT Resource**.
   b. In the Manage IT Resource page, click **Create IT Resource**.

4. 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.

5. Click **Continue**.

   - **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.

7. 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:** Oracle Fusion Middleware User’s Guide 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.

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.
4. Search for the rule name.

### 6.5.6 Reconciliation Action Rules for DB2

This section provides information on the reconciliation action rules for reconciliation. No action is performed for rule conditions that are not predefined for this connector. You can define your own action rule for such rule conditions. See Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for information about modifying or creating reconciliation action rules.

Table 6–13 lists the action rules for target resource reconciliation.

<table>
<thead>
<tr>
<th>Rule Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Matches Found</td>
<td>Assign to Administrator With Least Load</td>
</tr>
<tr>
<td>One Entity Match Found</td>
<td>Establish Link</td>
</tr>
<tr>
<td>One Process Match Found</td>
<td>Establish Link</td>
</tr>
</tbody>
</table>

Table 6–14 lists the action rules for trusted source reconciliation.

<table>
<thead>
<tr>
<th>Rule Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Matches Found</td>
<td>Create User</td>
</tr>
<tr>
<td>One Entity Match Found</td>
<td>Establish Link</td>
</tr>
</tbody>
</table>

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.
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.
6.5.7 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 Section 6.4.2, "Attributes for Scheduled Jobs for DB2" for information about this scheduled job.

6.5.8 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UID</strong></td>
<td>Unique identity representing the user This parameter is mapped to USERNAME or <strong>NAME</strong> connector attribute.</td>
</tr>
<tr>
<td>userType</td>
<td>Type of the user account The value of this parameter can be one of the following: USER or GROUP</td>
</tr>
</tbody>
</table>

For detailed information about ICF Filters, see the "ICF Filter Syntax" section of the Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager.

While deploying the connector, follow the instructions in Section 6.4.3, "Configuring Scheduled Jobs for DB2" to specify attribute values.

6.5.9 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 Section 6.4.3, "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.

See Also: The "Provisioning" section in Oracle Fusion Middleware User’s Guide for Oracle Identity Manager for conceptual information about provisioning.

This section contains the following topics about provisioning:

- Section 6.6.1, "Guidelines on Performing Provisioning Operations for DB2"
- Section 6.6.2, "Provisioning Process for DB2"
- Section 6.6.3, "Configuring Direct Provisioning for DB2"
- Section 6.6.4, "Configuring Request-Based Provisioning for DB2"
- Section 6.6.5, "Switching Between Request-Based Provisioning and Direct Provisioning for DB2"
- Section 6.6.6, "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 Provisioning Process for DB2

**See Also:** The "Provisioning" section in *Oracle Fusion Middleware User’s Guide for Oracle Identity Manager* for conceptual information about provisioning

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.

This 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 Section 6.6.5, "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: J Doe
- 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:

     \[ \text{GRANT USE OF TABLESPACE USERSPACE1 TO USER jdoe WITH GRANT OPTION} \]

   - If the administrator did not specify a value for granting tablespace With Grant Option, then the following SQL statement is formed:

     \[ \text{GRANT USE OF TABLESPACE USERSPACE1 TO USER jdoe} \]

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.
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.
      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 **DB2 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.
6.6.4 Configuring Request-Based Provisioning for DB2

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.

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

- Section 6.6.4.1, "Approver's Role in Request-Based Provisioning"
- Section 6.6.4.2, "Importing Request Datasets Using Deployment Manager"
- Section 6.6.4.3, "End User's Role in Request-Based Provisioning"
- Section 6.6.4.4, "Enabling the Auto Save Form Feature"
- Section 6.6.4.5, "Running the PurgeCache Utility"

### 6.6.4.1 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.

### 6.6.4.2 Importing Request Datasets Using Deployment Manager

**See Also:** Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager for detailed information about importing objects from an XML file using the 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.

### 6.6.4.3 End User’s Role in Request-Based Provisioning

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

**See Also:** *Oracle Fusion Middleware User’s Guide for Oracle Identity Manager* for detailed information about these steps

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
   - 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.

### 6.6.4.4 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 **DB2 DB** process definition.

4. Select the **Auto Save Form** check box.

5. Click the save icon.

### 6.6.4.5 Running the PurgeCache Utility

Run the PurgeCache utility to clear content belonging to the Metadata category from the server cache. See Section 2.3.1.3, "Clearing Content Related to 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

**Note:** It is assumed that you have performed the procedure described in Section 6.6.4, "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.

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

   **Note:** See the "Managing Users" chapter in *Oracle Fusion Middleware User’s Guide for 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 the "Managing Users" chapter in *Oracle Fusion Middleware User’s Guide for 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:

- Section 6.7.1, "Guidelines on Configuring the Queries for DB2"
- Section 6.7.2, "Configuring Queries to Add Support for Custom Parameters and Lookup Fields for DB2"
- Section 6.7.3, "Configuring the Connector for Multiple Installations of DB2"
- Section 6.7.4, "Configuring the Connector for Multiple Trusted Source Reconciliation from DB2"
- Section 6.7.5, "Configuring Validation of Data During Reconciliation and Provisioning for DB2"
- Section 6.7.6, "Configuring Transformation of Data During User Reconciliation for DB2"
- Section 6.7.7, "Configuring Resource Exclusion Lists for DB2"
- Section 6.7.8, "Configuring Action Scripts for DB2"

6.7.1 Guidelines on Configuring 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.

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

---

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

- **Section 6.7.1.1, "Syntax of Provisioning Queries for DB2 Database"**
- **Section 6.7.1.2, "Syntax of Reconciliation Queries for DB2 Database"**
- **Section 6.7.1.3, "Syntax of List of Values Queries for DB2 Database"**

### 6.7.1.1 Syntax of Provisioning Queries for DB2 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:

```sql
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: EXCLUDEVALIDATION"]
```
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_PENCED_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>
<thead>
<tr>
<th>Operation</th>
<th>Selection Logic</th>
<th>Query IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>CREATE_OBJECTTYPE</td>
<td>CREATE_USER</td>
</tr>
<tr>
<td>DELETE</td>
<td>DELETE_OBJECTTYPE</td>
<td>DELETE_USER</td>
</tr>
<tr>
<td>ENABLE</td>
<td>ENABLE_OBJECTTYPE</td>
<td>ENABLE_USER</td>
</tr>
<tr>
<td>DISABLE</td>
<td>DISABLE_OBJECTTYPE</td>
<td>DISABLE_USER</td>
</tr>
<tr>
<td>RESET PASSWORD</td>
<td>SET_PASSWORD</td>
<td>SET_PASSWORD</td>
</tr>
<tr>
<td>ADD CHILD VALUES</td>
<td>UPDATE_ADD_ATTRIBUTE</td>
<td>UPDATE_ADD_TABLESPACES</td>
</tr>
<tr>
<td>REMOVE CHILD VALUES</td>
<td>UPDATE_REVOKE_ATTRIBUTE</td>
<td>UPDATE_REVOKE_TABLESPACES</td>
</tr>
</tbody>
</table>

6.7.1.2 Syntax of Reconciliation Queries for DB2 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__}, {userType} FROM SYSIBM.SYSDBAUTH {filter}"
    QueryType="SQL"
    Parameters=["__UID__":"Type: String, Direction: OUT, ColName: GRANTEE",
              "userType":"Type: String, Direction: OUT, ColName: GRANTEEETYPE"
            ]
    QueryExtensions=["SEARCH_USER_TABLESPACE", "SEARCH_USER_SCHEMA", "SEARCH_USER_STATUS"]
}

In this syntax:

- QUERYID refers to the unique name of the query.

For example: SEARCH_USER

QUERYID can be one of the following values:
Extending the Connector for DB2

- 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 SYIBM.SYSDBAUTH (filter) SEARCH_USER_TABLESPACE, SEARCH_USER_SCHEMA, SEARCH_USER_STATUS

6.7.1.3 Syntax of List of Values Queries for DB2 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:

__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:

- Section 6.7.2.1, "Updating the Query Files for DB2 Database"
- Section 6.7.2.2, "Configuring Oracle Identity Manager"

6.7.2.1 Updating the Query Files for DB2 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:

   For Microsoft Windows:

   OIM_HOME/server/bin/DownloadJars.bat

   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 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.

**See Also:** *Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager* for detailed information about the Download JARs utility

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 Section 6.7.1, *Guidelines on Configuring the 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, CREATE TABLE, 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"]
   ```
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:

   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.

   **See Also:** Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager for detailed information about the Update JARs utility

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. See Step 2 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for more information.
   c. Create a new UI form to view the newly added field along with the rest of the fields. See Step 3 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for 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 4.c), and then save the application instance.
   e. Publish the sandbox. See Step 5 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for more information.

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:
b. Double-click **Process Definition** and open the **DB2 DB User** process definition.

c. In the process definition, add a new task for updating the field as follows:
   - Click **Add** and enter the task name, for example, **Custom Attribute 1 Updated**, and the task description.
   - In the Task Properties section, select the following fields:
     Conditional
     Allow Multiple Instances
   - Click the save icon.

d. On the Integration tab, click **Add**, and then click **Adapter**.

e. Select the **adpDB2UPDATECHILDTABLEVALUES** 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:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Map To</th>
<th>Qualifier</th>
<th>Literal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter return value</td>
<td>Object</td>
<td>Response code</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>attributeName</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>Custom Attribute 1</td>
</tr>
<tr>
<td>itRes</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_DB2_U_ITRES</td>
</tr>
<tr>
<td>objectType</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>User</td>
</tr>
<tr>
<td>processInstanceKey</td>
<td>Long</td>
<td>Process Data</td>
<td>Process Instance</td>
<td>NA</td>
</tr>
</tbody>
</table>

   g. On the Responses tab, click **Add** to add the following response codes:

<table>
<thead>
<tr>
<th>Code Name</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>Error occurred</td>
<td>R</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>An unknown response was received</td>
<td>R</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Operation completed</td>
<td>C</td>
</tr>
</tbody>
</table>

h. Click the save icon and then close the dialog box.

### 6.7.3 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.

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.

Table 6–16 Connector Objects and Their Associations

<table>
<thead>
<tr>
<th>Connector Object</th>
<th>Name</th>
<th>Referenced By</th>
<th>Comments on Creating a Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT resource</td>
<td>DB2 DB</td>
<td>UD_DB_DB2_U (process form)</td>
<td>Create a copy of the IT resource with a different name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scheduled tasks</td>
<td></td>
</tr>
<tr>
<td>Resource object</td>
<td>DB2 DB User</td>
<td>All connector operations</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>DB2 DB Trusted</td>
<td></td>
<td>Note: Create copies of the resource object only if there are differences in attributes between the various installations of the target system.</td>
</tr>
<tr>
<td>Scheduled Jobs</td>
<td>There are many scheduled jobs for different purposes.</td>
<td>NA</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for information on creating an application instance.
**Table 6–16 (Cont.) Connector Objects and Their Associations**

<table>
<thead>
<tr>
<th>Connector Object</th>
<th>Name</th>
<th>Referenced By</th>
<th>Comments on Creating a Copy</th>
</tr>
</thead>
</table>
| Process definition   | DB2 DB User           | NA                                 | 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.  
**Note:** Create copies of the process form only if there are differences in attributes between the various installations of the target system. |
| Process form          | UD_DB_DB2_U           | DB2 DB User (Process definition)   | 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.  
**Note:** Create copies of the process form only if there are differences in attributes between the various installations of the target system. |
| Child process form    | UD_DB_DB2_S           | DB2 DB User (Process definition)   | 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. |
| Child process form    | UD_DB_DB2_T           | UD_DB_DB2_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 | Lookup.DBUM.DB2_Configuration | DB2 (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. |
When you configure reconciliation:
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 perform provisioning operations:
When you use the Administrative and User Console to perform provisioning, you can specify the IT resource corresponding to the target system installation to which you want to provision the user.
6.7.4 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 Oracle Fusion Middleware Developer’s Guide for 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 (#):

   ```java
   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 Section 6.3.6, "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

9. In the Decode column, enter the name of the lookup you updated or created in
step 3.
   For example, Lookup.DBUM.DB2.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.

See Also: Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for detailed information about the Upload JARs utility

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.

### 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:

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:

   ```java
   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
           */
   ```

   ```java
   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 Section 6.3.5, "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.

See Also: Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for detailed information about the Upload JARs utility

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 Section 6.3.4, “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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>User001</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>User001</td>
</tr>
</tbody>
</table>

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/](http://download.oracle.com/javase/6/docs/api/java/util/regex/)

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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name[PATTERN]</td>
<td>User001</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name[PATTERN]</td>
<td>00012*</td>
</tr>
</tbody>
</table>

6. Click the save icon.

### 6.7.8 Configuring 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.

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.
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.

- Section 7.1, "Configuring Secure Communication Between Sybase and Oracle Identity Manager"
- Section 7.2, "Determining Values for the JDBC URL and Connection Properties Parameters for Sybase"
- Section 7.3, "Lookup Definitions for Sybase"
- Section 7.4, "Scheduled Jobs for Sybase"
- Section 7.5, "Reconciliation from Sybase"
- Section 7.6, "Provisioning for Sybase"
- Section 7.7, "Extending the Connector for Sybase"

### 7.1 Configuring Secure Communication Between Sybase and Oracle Identity Manager

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
-storespass 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.

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 Section 2.3.4, "Configuring the IT Resource for the Connector Server."

7.2 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 Section 2.3.2, "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:

```
```

- **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

Lookup definitions used during connector operations can be categorized as follows:

- **Section 7.3.1, "Lookup Definitions Synchronized with Sybase"**
- **Section 7.3.2, "Lookup Definitions for Configurations for Sybase"**
- **Section 7.3.3, "Lookup Definitions for Attribute Mappings for Sybase Login Entity"**
- **Section 7.3.4, "Lookup Definitions for Attribute Mappings for Sybase User Entity"**
- **Section 7.3.5, "Lookup Definition for Validation of Data in Sybase"**
- **Section 7.3.6, "Lookup Definitions for Transformation of Data in Sybase"**
- **Section 7.3.7, "Lookup Definitions for Exclusion Lists for Sybase"**

You must provide Decode values for some of the entries of the following lookup definitions. 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.

#### 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:** `ITRESOURCE_KEY~LOOKUP_FIELD_ID`
  - In this format:
    - `ITRESOURCE_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:** `ITRESOURCE_NAME~LOOKUP_FIELD_ID`
  - In this format:
    - `ITRESOURCE_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 on the Administrative and User Console, 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>
<thead>
<tr>
<th>Lookup Definition</th>
<th>Target Column Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup.DBUM.Sybase.Databases</td>
<td>Database Name</td>
</tr>
<tr>
<td>Lookup.DBUM.Sybase.DBGroups</td>
<td>Database Group</td>
</tr>
<tr>
<td>Lookup.DBUM.Sybase.DefaultLang</td>
<td>Default Language</td>
</tr>
<tr>
<td>Lookup.DBUM.Sybase.Roles</td>
<td>Role</td>
</tr>
</tbody>
</table>

### 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

- Section 7.3.2.1, "Lookup.DBUM.Sybase.Configuration"
- Section 7.3.2.2, "Lookup.DBUM.Sybase.Login.Configuration"
- Section 7.3.2.3, "Lookup.DBUM.Sybase.UM.Configuration"
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>
<thead>
<tr>
<th>Code Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bundle Name</td>
<td>org.identityconnectors.dbum</td>
</tr>
<tr>
<td></td>
<td>Name of the connector bundle package</td>
</tr>
<tr>
<td></td>
<td>Do not modify this entry.</td>
</tr>
<tr>
<td>Bundle Version</td>
<td>1.0.1116</td>
</tr>
<tr>
<td></td>
<td>Version of the connector bundle class</td>
</tr>
<tr>
<td></td>
<td>Do not modify this entry.</td>
</tr>
<tr>
<td>Connector Name</td>
<td>org.identityconnectors.dbum.DBUMConnector</td>
</tr>
<tr>
<td></td>
<td>Name of the connector class</td>
</tr>
<tr>
<td></td>
<td>Do not modify this entry.</td>
</tr>
<tr>
<td>disableValuesSet</td>
<td>&quot;YES&quot;</td>
</tr>
<tr>
<td></td>
<td>This entry is used internally. Do not modify this entry.</td>
</tr>
<tr>
<td>USERLOGIN Configuration Lookup</td>
<td>Lookup.DBUM.Sybase.Login.Configuration</td>
</tr>
<tr>
<td></td>
<td>Name of the lookup definition that contains configuration properties for login entities</td>
</tr>
<tr>
<td></td>
<td>Do not modify this entry.</td>
</tr>
<tr>
<td>User Configuration Lookup</td>
<td>Lookup.DBUM.Sybase.UM.Configur</td>
</tr>
<tr>
<td></td>
<td>Name of the lookup definition that contains user-specific configuration properties</td>
</tr>
<tr>
<td></td>
<td>Do not modify this entry.</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provisioning Attribute Map</td>
<td>Lookup.DBUM.Sybase.Login.ProvAttrMap</td>
<td></td>
</tr>
<tr>
<td>Provisioning Validation Lookup</td>
<td>Lookup.DBUM.Sybase.Login.ProvValidations</td>
<td></td>
</tr>
<tr>
<td>Recon Attribute Defaults</td>
<td>Lookup.DBUM.Sybase.Login.ReconDefaults</td>
<td></td>
</tr>
<tr>
<td>Recon Attribute Map</td>
<td>Lookup.DBUM.Sybase.Login.ReconAttrMap</td>
<td></td>
</tr>
<tr>
<td>Recon Transformation Lookup</td>
<td>Lookup.DBUM.Sybase.Login.ReconTransformations</td>
<td></td>
</tr>
</tbody>
</table>

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.
### 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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bundle Name</td>
<td>org.identityconnectors.dbum</td>
<td>This entry holds the name of the connector bundle package. Do not modify this entry.</td>
</tr>
<tr>
<td>Bundle Version</td>
<td>1.0.1116</td>
<td>This entry holds the version of the connector bundle class. Do not modify this entry.</td>
</tr>
<tr>
<td>Connector Name</td>
<td>org.identityconnectors.dbum.DBUMConnector</td>
<td>This entry holds the name of the connector class. Do not modify this entry.</td>
</tr>
<tr>
<td>disableValuesSet</td>
<td>&quot;YES&quot;</td>
<td>This entry is used internally. Do not modify this entry.</td>
</tr>
<tr>
<td>USERLOGIN Configuration Lookup</td>
<td>Lookup.DBUM.Sybase.UM.Configuration.Trusted</td>
<td>This entry holds the name of the lookup definition that contains login-specific configuration properties. Do not modify this entry.</td>
</tr>
</tbody>
</table>

### 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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recon Attribute Defaults</td>
<td>Lookup.DBUM.Sybase.UM.ReconDefaults.Trusted</td>
<td></td>
</tr>
<tr>
<td>Recon Attribute Map</td>
<td>Lookup.DBUM.Sybase.UM.ReconAttrMap.Trusted</td>
<td></td>
</tr>
<tr>
<td>Recon Exclusion List</td>
<td>Lookup.DBUM.Sybase.UM.ExclusionList.Trusted</td>
<td></td>
</tr>
<tr>
<td>Recon Transformation Lookup</td>
<td>Lookup.DBUM.Sybase.UM.ReconTransformations.Trusted</td>
<td></td>
</tr>
</tbody>
</table>
7.3.3 Lookup Definitions for Attribute Mappings for Sybase Login Entity

This section describes the following lookup definitions:

- Section 7.3.3.1, "Lookup.DBUM.Sybase.Login.ProvAttrMap"
- Section 7.3.3.2, "Lookup.DBUM.Sybase.Login.ReconAttrMap"
- Section 7.3.3.3, "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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Database[LOOKUP]</td>
<td>defaultDatabase</td>
</tr>
<tr>
<td>Default Language[LOOKUP]</td>
<td>defaultLanguage</td>
</tr>
<tr>
<td>Full Name</td>
<td>fullName</td>
</tr>
<tr>
<td>Password</td>
<td><strong>PASSWORD</strong></td>
</tr>
<tr>
<td>Login Name</td>
<td><strong>NAME</strong></td>
</tr>
<tr>
<td>Return ID</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>UD_DB_SYB_R~Role[LOOKUP]</td>
<td>roleList<del>role</del><strong>NAME</strong></td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Database Name[LOOKUP]</td>
<td>defaultDatabase</td>
</tr>
<tr>
<td>Default Language[LOOKUP]</td>
<td>defaultLanguage</td>
</tr>
<tr>
<td>Full Name</td>
<td>fullName</td>
</tr>
<tr>
<td>Login Name</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>Reference ID</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>Roles List~Role Name[LOOKUP]</td>
<td>roleList</td>
</tr>
<tr>
<td>Status</td>
<td><strong>ENABLE</strong></td>
</tr>
</tbody>
</table>

Table 7-7 (Cont.) Entries in Lookup.DBUM.Sybase.UM.Configuration.Trusted

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Dcode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recon Validation Lookup</td>
<td>Lookup.DBUM.Sybase.UM.ReconValidation.Trusted</td>
</tr>
</tbody>
</table>
7.3.3  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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

7.3.4  Lookup Definitions for Attribute Mappings for Sybase User Entity
This section describes the following lookup definitions:

- Section 7.3.4.1, "Lookup.DBUM.Sybase.UM.ProvAttrMap"
- Section 7.3.4.2, "Lookup.DBUM.Sybase.UM.ReconAttrMap"
- Section 7.3.4.3, "Lookup.DBUM.Sybase.UM.ReconDefaults"
- Section 7.3.4.4, "Lookup.DBUM.Sybase.UM.ReconDefaults.Trusted"
- Section 7.3.4.5, "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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Group[LOOKUP]</td>
<td>databaseGroup</td>
</tr>
<tr>
<td>Database Name</td>
<td>databaseName</td>
</tr>
<tr>
<td>Login Name</td>
<td>loginName</td>
</tr>
<tr>
<td>Return ID</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>Username</td>
<td><strong>NAME</strong></td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Group</td>
<td>databaseGroup</td>
</tr>
<tr>
<td>Login Name</td>
<td>loginName</td>
</tr>
<tr>
<td>Reference ID</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>User Name</td>
<td><strong>UID</strong></td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

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 lists the default entries in this lookup definition.

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empl Type</td>
<td>Full-Time</td>
</tr>
<tr>
<td>Organization Name</td>
<td>Xellerate Users</td>
</tr>
<tr>
<td>Status</td>
<td>Active</td>
</tr>
<tr>
<td>User Type</td>
<td>End-User</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Name</td>
<td>fullName</td>
</tr>
<tr>
<td>First Name</td>
<td><strong>UID</strong></td>
</tr>
<tr>
<td>Status</td>
<td><strong>ENABLE</strong></td>
</tr>
<tr>
<td>User ID</td>
<td><strong>UID</strong></td>
</tr>
</tbody>
</table>

7.3.5 Lookup Definition for Validation of Data in Sybase

Section 7.7.5, "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>
<thead>
<tr>
<th>Code Key</th>
<th>Decode Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roles List</td>
<td>oracle.iam.connectors.dbum.transformations.SybaseRoleTransformation</td>
</tr>
</tbody>
</table>

- For Sybase user entity: Lookup.DBUM.Sybase.UM.ReconTransformations

- For Sybase user entity in trusted source mode:
  Lookup.DBUM.Sybase.UM.ReconTransformations.Trusted

Section 7.7.6, "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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
<th>Sample Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Name</td>
<td>User ID of a user</td>
<td>Code Key: Login Name&lt;br&gt;Decide: User001</td>
</tr>
<tr>
<td>Login Name with the [PATTERN] suffix</td>
<td>A regular expression supported by the representation in the Java.util.regex.Pattern class</td>
<td>Code Key: Login Name[PATTERN]&lt;br&gt;To exclude users matching any of the user ID’s User001, User002, User088, then:&lt;br&gt;Decide: User001</td>
</tr>
</tbody>
</table>

See Also: For information about the supported patterns, visit http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html

Section 7.7.7, "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:
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>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Key Attribute</td>
<td>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: <strong>NAME</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Do not change the value of this attribute.</td>
</tr>
<tr>
<td>Decode Attribute</td>
<td>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: <strong>NAME</strong></td>
</tr>
<tr>
<td>IT Resource Name</td>
<td>Enter the name of the IT resource for the target system installation from which you want to reconcile user records. Default value: Sybase DB</td>
</tr>
</tbody>
</table>

Table 7–17  Attributes of the Scheduled Jobs for Lookup Field Synchronization
### 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 7-17 (Cont.) Attributes of the Scheduled Jobs for Lookup Field Synchronization**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lookup Name</strong></td>
<td>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:</td>
</tr>
<tr>
<td>DBUM Sybase Databases Lookup Reconciliation - Lookup.DBUM.Sybase.Databases</td>
<td></td>
</tr>
<tr>
<td>DBUM Sybase DB Groups Lookup Reconciliation - Lookup.DBUM.Sybase.DBGroups</td>
<td></td>
</tr>
<tr>
<td>DBUM Sybase Languages Lookup Reconciliation - Lookup.DBUM.Sybase.DefaultLang</td>
<td></td>
</tr>
<tr>
<td>DBUM Sybase Roles Lookup Reconciliation - Lookup.DBUM.Sybase.Roles</td>
<td></td>
</tr>
<tr>
<td><strong>Object Type</strong></td>
<td>Enter the type of object whose values must be synchronized. Depending on the scheduled job you are using, the default values are as follows:</td>
</tr>
<tr>
<td>DBUM Sybase Databases Lookup Reconciliation - <strong>DATABASES</strong></td>
<td></td>
</tr>
<tr>
<td>DBUM Sybase DB Groups Lookup Reconciliation - <strong>DBGROUPS</strong></td>
<td></td>
</tr>
<tr>
<td>DBUM Sybase Languages Lookup Reconciliation - <strong>DEFAULTLANG</strong></td>
<td></td>
</tr>
<tr>
<td>DBUM Sybase Roles Lookup Reconciliation - <strong>ROLES</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> Do not change the value of this attribute.</td>
<td></td>
</tr>
<tr>
<td><strong>Resource Object Name</strong></td>
<td>Enter the name of the resource object that is used for reconciliation.</td>
</tr>
<tr>
<td>Default value: Sybase DB User</td>
<td></td>
</tr>
</tbody>
</table>
### Table 7–18 Attributes of the Scheduled Jobs for Reconciliation

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT Resource Name</td>
<td>Name of the IT resource for the target system installation from which you want to reconcile user records</td>
</tr>
<tr>
<td></td>
<td>Default value: Sybase DB</td>
</tr>
<tr>
<td></td>
<td>For DBUM Sybase Trusted Reconciliation, enter the name of the IT resource created for trusted source mode.</td>
</tr>
<tr>
<td>Object Type</td>
<td>Type of object you want to reconcile</td>
</tr>
<tr>
<td></td>
<td>■ For DBUM Sybase User Target Reconciliation: User</td>
</tr>
<tr>
<td></td>
<td>■ For DBUM Sybase User Login Target Reconciliation and DBUM Sybase Trusted Reconciliation: USERLOGIN</td>
</tr>
<tr>
<td>Resource Object Name</td>
<td>Name of the resource object that is used for reconciliation</td>
</tr>
<tr>
<td></td>
<td>■ For DBUM Sybase User Target Reconciliation: Sybase DB User</td>
</tr>
<tr>
<td></td>
<td>■ For DBUM Sybase User Login Target Reconciliation: Sybase DB User Login</td>
</tr>
<tr>
<td></td>
<td>■ For DBUM Sybase Trusted Reconciliation: Sybase UserLogin Trusted</td>
</tr>
<tr>
<td>Scheduled Task Name</td>
<td>Name of the scheduled job</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> 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.</td>
</tr>
</tbody>
</table>

Table 7–19 describes the attributes of the scheduled jobs for delete operations.

### Table 7–19 Attributes of the Scheduled Jobs for Delete Operations

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT Resource Name</td>
<td>Name of the IT resource for the target system installation from which you want to reconcile user records</td>
</tr>
<tr>
<td></td>
<td>Default value: Sybase DB</td>
</tr>
<tr>
<td></td>
<td>For DBUM Sybase Delete Trusted Reconciliation, enter the name of the IT resource created for trusted source mode.</td>
</tr>
<tr>
<td>Object Type</td>
<td>Type of object you want to reconcile</td>
</tr>
<tr>
<td></td>
<td>■ For DBUM Sybase User Target Reconciliation: User</td>
</tr>
<tr>
<td></td>
<td>■ For DBUM Sybase User Login Target Reconciliation and DBUM Sybase Trusted Reconciliation: USERLOGIN</td>
</tr>
<tr>
<td>Resource Object Name</td>
<td>Name of the resource object that is used for reconciliation</td>
</tr>
<tr>
<td></td>
<td>■ For DBUM Sybase Delete User Target Reconciliation: Sybase DB User</td>
</tr>
<tr>
<td></td>
<td>■ For DBUM Sybase Delete User Login Target Reconciliation: Sybase DB User Login</td>
</tr>
<tr>
<td></td>
<td>■ For DBUM Sybase Delete Trusted Reconciliation: Sybase UserLogin Trusted</td>
</tr>
</tbody>
</table>

**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 Section 7.4.1, "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. Depending on the Oracle Identity Manager release you are using, perform one of the following steps:
   - 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.
   - For 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.

2. Search for and open the scheduled job as follows:
   - 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.
   - In the search results table on the left pane, click the scheduled job in the Job Name column.

3. 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 Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager for detailed information about schedule types.

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

4. 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.

5. After specifying the attributes, click Apply to save the changes.

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

### 7.5 Reconciliation from Sybase

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:

- Section 7.5.1, "Reconciliation Process for Sybase"
- Section 7.5.2, "Reconciliation Rules for Sybase"
- Section 7.5.3, "Reconciliation Action Rules for Sybase"
- Section 7.5.4, "Performing Full Reconciliation from Sybase"

### 7.5.1 Reconciliation Process for Sybase

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

**See Also:** The "Reconciliation" section in *Oracle Fusion Middleware User’s Guide for 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.

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. With the target system as a target resource:
   
   a. 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. See Section 7.5.2, "Reconciliation Rules for Sybase" for information about the reconciliation rule.

   b. 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.

### 7.5.2 Reconciliation Rules for Sybase

**See Also:** Oracle Fusion Middleware User’s Guide 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.

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.
4. Search for the rule name.

### 7.5.3 Reconciliation Action Rules for Sybase

This section provides information on the reconciliation action rules for reconciliation.

No action is performed for rule conditions that are not predefined for this connector. You can define your own action rule for such rule conditions. See Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for information about modifying or creating reconciliation action rules.

Table 7-20 lists the action rules for target resource reconciliation.

<table>
<thead>
<tr>
<th>Rule Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Matches Found</td>
<td>Assign to Administrator With Least Load</td>
</tr>
<tr>
<td>One Entity Match Found</td>
<td>Establish Link</td>
</tr>
<tr>
<td>One Process Match Found</td>
<td>Establish Link</td>
</tr>
</tbody>
</table>

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.
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.

### 7.5.4 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 Section 7.4.2, "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.

**See Also:** The "Provisioning" section in Oracle Fusion Middleware User’s Guide for Oracle Identity Manager for conceptual information about provisioning.

This section contains the following topics about provisioning:
- Section 7.6.1, "Guidelines on Performing Provisioning Operations for Sybase"
- Section 7.6.2, "Provisioning Process for Sybase"
- Section 7.6.3, "Configuring Direct Provisioning for Sybase"
- Section 7.6.4, "Configuring Request-Based Provisioning for Sybase"
- Section 7.6.5, "Switching Between Request-Based Provisioning and Direct Provisioning for Sybase"
- Section 7.6.6, "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 Provisioning Process for Sybase

See Also: The "Provisioning" section in Oracle Fusion Middleware User’s Guide for Oracle Identity Manager for conceptual information about provisioning

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.

This 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 Section 7.6.5, "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: JDoee
- 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: JDoee
- 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 jdoee 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 jdoee 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.


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

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.

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

- Section 7.6.4.1, "Approver's Role in Request-Based Provisioning"
- Section 7.6.4.2, "Importing Request Datasets Using Deployment Manager"
- Section 7.6.4.3, "End User's Role in Request-Based Provisioning"
- Section 7.6.4.4, "Enabling the Auto Save Form Feature"
- Section 7.6.4.5, "Running the PurgeCache Utility"

#### 7.6.4.1 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 Importing Request Datasets Using Deployment Manager

See Also: Oracle Fusion Middleware Administrator’s Guide for Oracle Identity Manager for detailed information about importing objects from an XML file using the 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.

7.6.4.3 End User’s Role in Request-Based Provisioning

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

See Also: Oracle Fusion Middleware User’s Guide for Oracle Identity Manager for detailed information about these steps

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.

### 7.6.4.4 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.

### 7.6.4.5 Running the PurgeCache Utility

Run the PurgeCache utility to clear content belonging to the Metadata category from the server cache. See Section 2.3.1.3, "Clearing Content Related to 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

**Note:** It is assumed that you have performed the procedure described in Section 7.6.4, "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.

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

   Note: See the "Managing Users" chapter in Oracle Fusion Middleware User's Guide for 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:

   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:

- **Section 7.7.1**, "Guidelines on Configuring the Queries for Sybase"
- **Section 7.7.2**, "Configuring Queries to Add Support for Custom Parameters and Lookup Fields for Sybase"
- **Section 7.7.3**, "Configuring the Connector for Multiple Installations of Sybase"
- **Section 7.7.4**, "Configuring the Connector for Multiple Trusted Source Reconciliation from Sybase"
- **Section 7.7.5**, "Configuring Validation of Data During Reconciliation and Provisioning for Sybase"

---

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

---

**Note:** See the "Managing Users" chapter in *Oracle Fusion Middleware User’s Guide for Oracle Identity Manager* for more information about searching a user.
7.7 Guidelines on Configuring 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.

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

- Section 7.7.1.1, "Syntax of Provisioning Queries for Sybase Database"
- Section 7.7.1.2, "Syntax of Reconciliation Queries for Sybase Database"
- Section 7.7.1.3, "Syntax of List of Values Queries for Sybase Database"

### 7.7.1.1 Syntax of Provisioning Queries for Sybase Database

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

```plaintext
QUERYID { Query="QUERY" QueryType="QUERYTYPE" Parameters=["PARAM1":"PARAMDEF1", "PARAM2":"PARAMDEF2"...] 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 \textit{EXTENSION1}, \textit{EXTENSION2}, and so on.

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

\begin{verbatim}
QUERY PARAM1, PARAM2... [EXTENSIONJOIN [EXTENSION1
EXTENSIONSEPARATOR EXTENSION2 EXTENSIONSEPARATOR...]]
\end{verbatim}

For example:

\begin{verbatim}
CALL sp_adduser {loginName}, {__NAME__}, {databaseGroup}
\end{verbatim}

Table 7–21 lists the script selection logic of the provisioning queries:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Selection Logic</th>
<th>Query IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>CREATE_OBJECTTYPE</td>
<td>CREATE_USER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CREATE_USERLOGIN</td>
</tr>
<tr>
<td>DELETE</td>
<td>DELETE_OBJECTTTYPE</td>
<td>DELETE_USER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DELETE_USERLOGIN</td>
</tr>
<tr>
<td>ENABLE</td>
<td>ENABLE_OBJECTTTYPE</td>
<td>ENABLE_USERLOGIN</td>
</tr>
<tr>
<td>DISABLE</td>
<td>DISABLE_OBJECTTTYPE</td>
<td>DISABLE_USERLOGIN</td>
</tr>
<tr>
<td>RESET PASSWORD</td>
<td>OBJECTTTYPE_SET_PASSWORD</td>
<td>USERLOGIN_SET_PASSWORD</td>
</tr>
<tr>
<td>ADD CHILD VALUES</td>
<td>OBJECTTTYPE_UPDATE_ADD_</td>
<td>USERLOGIN_UPDATE_ADD_ROLELIST</td>
</tr>
<tr>
<td></td>
<td>ATTRIBUTE</td>
<td></td>
</tr>
<tr>
<td>REMOVE CHILD VALUES</td>
<td>OBJECTTTYPE_UPDATE_ REVOKE_ATTRIBUTE</td>
<td>USERLOGIN_UPDATE_REVOKE_ROLELIST</td>
</tr>
<tr>
<td>UPDATE</td>
<td>OBJECTTTYPE_UPDATE_</td>
<td>USERLOGIN_UPDATE_DEFAULTLANGUAGE</td>
</tr>
<tr>
<td></td>
<td>ATTRIBUTE</td>
<td>USERLOGIN_UPDATE_DEFAULTDATABASE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>USERLOGIN_UPDATE_DEFAULTDATABASE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UPDATE_DATABASEGROUP</td>
</tr>
</tbody>
</table>

7.7.1.2 Syntax of Reconciliation Queries for Sybase Database

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

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

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:

\[
\text{QUERY PARAM1, PARAM2... [EXTENSIONJOIN [EXTENSION1 EXTENSIONSEPARATOR EXTENSION2 EXTENSIONSEPARATOR...]]}
\]

For example:

\[
call \text{sp\_helpuser}() \{__UID__\}, \{\text{databaseGroup}\}, \{\text{loginName}\}
\]

### 7.7.1.3 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:

\[
\text{OBJECTTYPE} = \text{"QUERY"}
\]

For example:

\[
\text{__DEFAULTLANG__} = \text{"SELECT alias FROM syslanguages"}
\]

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: \text{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:

- **Section 7.7.2.1, "Updating the Query Files for Sybase Database"**
- **Section 7.7.2.2, "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:
For Microsoft Windows:

\textit{OIM\_HOME/server/bin/DownloadJars.bat}

For UNIX:

\textit{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.

\textbf{See Also:} \textit{Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager} for detailed information about the Download JARs utility

2. Copy the bundle JAR file in a temporary directory.

   \textbf{Sample JAR file:} bundle/org.identityconnectors.dbum-1.0.1116.jar

   \textbf{Sample temporary directory:} c:\temp

3. Run the following command to extract the connector bundle JAR file:

   \texttt{jar -xvf org.identityconnectors.dbum-1.0.1116.jar}

\textbf{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 \texttt{ConnectorBundle-Version} in the manifest file, META-INF/MANIFEST.MF, to a new value.

   \textbf{For example:}

   \texttt{ConnectorBundle-Version: 1.0.1117}

6. Depending on your requirement, update the query files with new parameters as per the query syntax described in Section 7.7.1, “Guidelines on Configuring the Queries for Sybase.”

   \textbf{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.

      \textbf{Sample query file:}

      \texttt{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', 'databaseGroup': 'Type: String',
    '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:

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.

See Also: Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager for detailed information about the Update JARs utility

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. See Step 2 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for more information.
   c. Create a new UI form to view the newly added field along with the rest of the fields. See Step 3 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for 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 4.c), and then save the application instance.
   e. Publish the sandbox. See Step 5 of Section 2.3.1.7, "Configuring Oracle Identity Manager Release 11.1.2 or Later" for more information.

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:


b. Double-click Process Definition and open the Sybase DB User process definition.

c. In the process definition, add a new task for updating the field as follows:
   – Click Add and enter the task name, for example, Custom Attribute 1 Updated, and the task description.
   – In the Task Properties section, select the following fields:
     Conditional
     Allow Multiple Instances
   – Click the save icon.

d. On the Integration tab, click Add, and then click Adapter.

e. Select the adpSYBASEDBUMUPDATEUSER 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:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Map To</th>
<th>Qualifier</th>
<th>Literal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter return value</td>
<td>Object</td>
<td>Response code</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>attributeName</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>Custom Attribute 1</td>
</tr>
<tr>
<td>itRes</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_SYB_U_ITRES</td>
</tr>
<tr>
<td>objectType</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>User</td>
</tr>
<tr>
<td>processInstanceKey</td>
<td>Long</td>
<td>Process Data</td>
<td>Process Instance</td>
<td>NA</td>
</tr>
</tbody>
</table>


g. On the Responses tab, click Add to add the following response codes:

<table>
<thead>
<tr>
<th>Code Name</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>Error occurred</td>
<td>R</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>An unknown response was received</td>
<td>R</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Operation completed</td>
<td>C</td>
</tr>
</tbody>
</table>

h. Click the save icon and then close the dialog box.

7.7.3 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.

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 Section 2.3.1.7, "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

<table>
<thead>
<tr>
<th>Connector Object</th>
<th>Name</th>
<th>Referenced By</th>
<th>Comments on Creating a Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT resource</td>
<td>Sybase DB</td>
<td>• UD_DB_SYB_U (process form)</td>
<td>Create a copy of the IT resource with a different name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Scheduled tasks</td>
<td></td>
</tr>
<tr>
<td>Resource object</td>
<td>Sybase DB User</td>
<td>All connector operations</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Sybase DB Trusted</td>
<td></td>
<td><strong>Note:</strong> Create copies of the resource object only if there are differences in attributes between the various installations of the target system.</td>
</tr>
<tr>
<td>Scheduled Jobs</td>
<td>There are many</td>
<td>NA</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>scheduled jobs</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>for different purposes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process definition</td>
<td>Sybase DB User</td>
<td>NA</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> Create copies of the process form only if there are differences in attributes between the various installations of the target system.</td>
</tr>
<tr>
<td>Process form</td>
<td>UD_DB_SYB_U</td>
<td>Sybase DB User and Sybase DB User Login (Process definitions)</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>UD_DB_SYB_L</td>
<td></td>
<td><strong>Note:</strong> Create copies of the process form only if there are differences in attributes between the various installations of the target system.</td>
</tr>
</tbody>
</table>
When you configure reconciliation:
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 perform provisioning operations:
When you use the Administrative and User Console to perform provisioning, you can specify the IT resource corresponding to the target system installation to which you want to provision the user.
7.7.4 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.

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 Section 7.3.5, "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.

    **See Also:** *Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager* for detailed information about the Upload JARs utility

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`.

---

7-40 Oracle Identity Manager Connector Guide for Database User Management
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:

```java
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 Section 7.3.6, "Lookup Definitions for Transformation of Data in Sybase."

For example, `

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.DBUMTransfomation`

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.

See Also: Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for detailed information about the Upload JARs utility

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 Section 7.3.7, "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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Name</td>
<td>User001</td>
</tr>
</tbody>
</table>

   **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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Name</td>
<td>User001</td>
</tr>
</tbody>
</table>

   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](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:
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:

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Name[PATTERN]</td>
<td>User001</td>
</tr>
</tbody>
</table>

6. Click the save icon.

### 7.7.8 Configuring 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.

<table>
<thead>
<tr>
<th>Code Key</th>
<th>Decode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Name[PATTERN]</td>
<td>00012*</td>
</tr>
</tbody>
</table>

**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.

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:
     ```sql
     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.
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:

- Section 8.1, “Target System Attributes and Queries”
- Section 8.2, “Configuring the Queries”
- Section 8.3, “Updating the Query Files to the Connector Bundle”
- Section 8.4, “Configuring the IT Resource”
- Section 8.5, “Configuring the Process Form”
- Section 8.6, “Configuring the Resource Object”
- Section 8.7, “Adding Process Tasks, Assigning Adapters, and Mapping Adapter Variables”
- Section 8.8, “Adding Attributes for Reconciliation”
- Section 8.9, “Configuring Lookup Definitions Used During Connector Operations”
- Section 8.10, “Configuring Scheduled Jobs”
- Section 8.11, “Configuring Oracle Identity Manager for Request-Based Provisioning”
- Section 8.12, “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:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type of Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Name</td>
<td>String</td>
</tr>
<tr>
<td>User Password</td>
<td>String</td>
</tr>
<tr>
<td>Database ID</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>List of values available in the DBNames table.</td>
</tr>
<tr>
<td>Status</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>Sample values: ACTIVE, DISABLED</td>
</tr>
<tr>
<td>lastModifiedToken</td>
<td>Long</td>
</tr>
</tbody>
</table>

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.
Create a new file called `Provisioning.queries` and add the following queries in the file:

```objectivec
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:GuardedString"]
    QueryExtensions=[]
}

UPDATE_DBID {
    Query="call UPDATE_DBID({__UID__}, {dbid})"
    QueryType="STOREDPROC"
    Parameters=["__UID__":"Type:String","dbid":"Type:String"]
    QueryExtensions=[]
}
```

Create a new file called `LoVSearch.queries` and add the following query in the file:

```objectivec
__DBNAME__=" Select id from DBNames"
```

Create a new file called `Search.queries` and add the following queries in the file:

```objectivec
SEARCH_USER {
    Query="SELECT {__UID__}, {dbid}, {status}, {lastModified} FROM MYDBUsers"
    QueryType="STOREDPROC"
    Parameters=["__UID__":"Type:String"]
    QueryExtensions=[]
}
```

**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:

- Section 4.7.1, "Guidelines on Configuring the Queries for Oracle Database"
- Section 3.7.1, "Guidelines on Configuring the Queries for MSSQL"
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.
   
   See Also: Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for detailed information about the Download JARs utility
   ```

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:
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:
   - 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.

   **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.

   **See Also:** Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager for detailed information about the Update JARs utility

7. If the connector is not installed, then perform the procedure specified in the following sections to deploy the connector:
   - **Section 2.1, "Preinstallation"
   - **Section 2.2, "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:** The "Managing Connector Lifecycle" chapter of Oracle Fusion Middleware Administrator’s Guide for 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. Depending on the Oracle Identity Manager release you are using, perform one of the following steps:
   ■ 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 Configuration region, click Manage IT Resource.
   ■ For 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 the "Managing Sandboxes" section of Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager.
     c. In the left pane, under Configuration, click IT Resource.

2. 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

3. Click the edit icon for the IT resource.

4. From the list at the top of the page, select Details and Parameters.

5. 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>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Type</td>
<td>This field identifies database type (such as Oracle and MSSQL) and its used for loading respective scripts. Sample value: mydb</td>
</tr>
<tr>
<td>JDBC Driver</td>
<td>Specify the value of the JDBC driver class name for MyDatabase.</td>
</tr>
<tr>
<td>JDBC URL</td>
<td>Specify the JDBC URL for MyDatabase.</td>
</tr>
<tr>
<td>Login Password</td>
<td>Enter the password for the user name of the MyDatabase account to be used for connector operations.</td>
</tr>
<tr>
<td>Login User</td>
<td>Enter the user name of the MyDatabase account to be used for connector operations.</td>
</tr>
</tbody>
</table>

6. 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.

4. Add the new fields for the MyDatabase user attributes on the process form.
   a. Click Add.
      A field is added to the list. Enter the details of the field.
   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:

A sample screenshot for the Database ID attribute is as follows:
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.

A sample screenshot of the updated process task is as follows:

---

To integrate the Create User process task with the adpORACREATESETFORM adapter:

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:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Map To</th>
<th>Qualifier</th>
<th>Literal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter return value</td>
<td>Object</td>
<td>Literal</td>
<td>NA</td>
<td>Response Code</td>
</tr>
<tr>
<td>itRes</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_ORA_U_ITRES</td>
</tr>
<tr>
<td>objectType</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>User</td>
</tr>
<tr>
<td>processInstanceKey</td>
<td>Long</td>
<td>Process Data</td>
<td>Process Instance</td>
<td>NA</td>
</tr>
<tr>
<td>shouldUpdateFlag</td>
<td>String</td>
<td>Literal</td>
<td>NA</td>
<td>False</td>
</tr>
<tr>
<td>updateField</td>
<td>String</td>
<td>Literal</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>updateValue</td>
<td>String</td>
<td>Literal</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

7. On the Responses tab, click Add to add the following response codes:

<table>
<thead>
<tr>
<th>Code Name</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>Error occurred</td>
<td>R</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>An unknown response was received</td>
<td>R</td>
</tr>
<tr>
<td>SUCCESS</td>
<td>Operation completed</td>
<td>C</td>
</tr>
</tbody>
</table>

8. 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:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Map To</th>
<th>Qualifier</th>
<th>Literal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter return value</td>
<td>Object</td>
<td>Literal</td>
<td>NA</td>
<td>Response Code</td>
</tr>
<tr>
<td>itRes</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_ORA_U_ITRES</td>
</tr>
<tr>
<td>objectType</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>User</td>
</tr>
<tr>
<td>processInstanceKey</td>
<td>Long</td>
<td>Process Data</td>
<td>Process Instance</td>
<td>NA</td>
</tr>
<tr>
<td>shouldUpdateFlag</td>
<td>String</td>
<td>Literal</td>
<td>NA</td>
<td>False</td>
</tr>
<tr>
<td>updateField</td>
<td>String</td>
<td>Literal</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>updateValue</td>
<td>String</td>
<td>Literal</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>
Using the procedure described previously, integrate the **Disable User** process task with the `adpORADISABLEUPDATEFORM` adapter and map the following adapter variables:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Map To</th>
<th>Qualifier</th>
<th>Literal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter return value</td>
<td>Object</td>
<td>Literal</td>
<td>NA</td>
<td>Response Code</td>
</tr>
<tr>
<td>itRes</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_ORA_U_ITRES</td>
</tr>
<tr>
<td>objectType</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>User</td>
</tr>
<tr>
<td>processInstanceKey</td>
<td>Long</td>
<td>Process Data</td>
<td>Process Instance</td>
<td>NA</td>
</tr>
<tr>
<td>shouldUpdateFlag</td>
<td>String</td>
<td>Literal</td>
<td>NA</td>
<td>False</td>
</tr>
<tr>
<td>updateField</td>
<td>String</td>
<td>Literal</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>updateValue</td>
<td>String</td>
<td>Literal</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

Using the procedure described previously, integrate the **Delete User** process task with the `adpORADELETEUSER` adapter and map the following adapter variables:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Map To</th>
<th>Qualifier</th>
<th>Literal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter return value</td>
<td>Object</td>
<td>Literal</td>
<td>NA</td>
<td>Response Code</td>
</tr>
<tr>
<td>itResFieldName</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_ORA_U_ITRES</td>
</tr>
<tr>
<td>objectType</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>User</td>
</tr>
<tr>
<td>processInstanceKey</td>
<td>Long</td>
<td>Process Data</td>
<td>Process Instance</td>
<td>NA</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Map To</th>
<th>Qualifier</th>
<th>Literal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter return value</td>
<td>Object</td>
<td>Literal</td>
<td>NA</td>
<td>Response Code</td>
</tr>
<tr>
<td>attrName</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_ORA_U_PASSWORD</td>
</tr>
<tr>
<td>ITResField</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_ORA_U_ITRES</td>
</tr>
<tr>
<td>objectType</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>User</td>
</tr>
<tr>
<td>processInstanceKey</td>
<td>Long</td>
<td>Process Data</td>
<td>Process Instance</td>
<td>NA</td>
</tr>
<tr>
<td>oldValue</td>
<td>String</td>
<td>Process Data</td>
<td>NA</td>
<td>Password (Field with old value box checked)</td>
</tr>
<tr>
<td>newValue</td>
<td>String</td>
<td>Process Data</td>
<td>NA</td>
<td>Password (Field with old value box unchecked)</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Data Type</th>
<th>Map To</th>
<th>Qualifier</th>
<th>Literal Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter return value</td>
<td>Object</td>
<td>Literal</td>
<td>NA</td>
<td>Response Code</td>
</tr>
<tr>
<td>attrName</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_ORA_U_DBID</td>
</tr>
<tr>
<td>itResFieldName</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>UD_DB_ORA_U_ITRES</td>
</tr>
<tr>
<td>objectType</td>
<td>String</td>
<td>Literal</td>
<td>String</td>
<td>User</td>
</tr>
<tr>
<td>processInstanceKey</td>
<td>Long</td>
<td>Process Data</td>
<td>Process Instance</td>
<td>NA</td>
</tr>
<tr>
<td>oldValue</td>
<td>Long</td>
<td>Process Data</td>
<td>Process Instance</td>
<td>Database ID (Field withhold Value box checked)</td>
</tr>
<tr>
<td>newValue</td>
<td>Long</td>
<td>Process Data</td>
<td>Process Instance</td>
<td>Database ID (Field withhold Value box unchecked)</td>
</tr>
</tbody>
</table>

### 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:

![Attribute Mappings Screenshot](image)

### 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 a lookup definition:

1. Log in to the Design Console.
2. Expand Administration, and then double-click Lookup Definition.
3. Search for and open the lookup definition that you want to modify.
   For example: Lookup.DBUM.Oracle.Configuration
4. Enter the value in the Decode column for the Code Key that you want to set.
5. Click the save icon.

By performing the procedure described in this section, you must update the following lookup definitions:

- **Lookup.DBUM.Oracle.Configuration**
  Update the Decode column of disabledValuesSet to Disabled. 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.DBUM.Oracle.UM.ProvAttrMap**
  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). 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:
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.

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:

**See Also:** Section 4.3, "Lookup Definitions 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:

- **Section 8.10.1, "Configuring Scheduled Jobs for Lookup Field Synchronization"**
  For synchronizing lookup field values with the target system

- **Section 8.10.2, "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 Section 4.4.1, "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:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Key Attribute</td>
<td>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: <strong><strong>NAME</strong></strong></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Do not change the value of this attribute.</td>
</tr>
<tr>
<td>Decode Attribute</td>
<td>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: <strong><strong>NAME</strong></strong></td>
</tr>
<tr>
<td>IT Resource Name</td>
<td>Enter the name of the IT resource for the target system installation from which you want to reconcile user records. For example: <strong>MYDB</strong></td>
</tr>
<tr>
<td>Lookup Name</td>
<td>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: <strong>Lookup.DBUM.MYDB.DBNames</strong></td>
</tr>
<tr>
<td>Object Type</td>
<td>Enter the type of object whose values must be synchronized. For example: <strong><strong>DBNAME</strong></strong> (as specified in the LoVSearch.queries file)</td>
</tr>
<tr>
<td>Resource Object Name</td>
<td>Enter the name of the resource object that is used for reconciliation. For example: <strong>MYDB User</strong></td>
</tr>
</tbody>
</table>

8.10.2 Configuring Scheduled Jobs for Reconciliation

To fetching data from the target system for reconciliation with Oracle Identity Manager, update the **DBUM Oracle User Target Reconciliation** scheduled job parameters as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT Resource Name</td>
<td>Enter the name of the IT resource for the target system installation from which you want to reconcile user records. For example: <strong>MYDB</strong></td>
</tr>
<tr>
<td>Object Type</td>
<td>Enter the type of object whose values must be synchronized. For example: <strong>User</strong></td>
</tr>
</tbody>
</table>
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.

Similarly, you can update the **DBUM Oracle Delete User Target Reconciliation** scheduled job parameters as follows:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IT Resource Name</strong></td>
<td>Enter the name of the IT resource for the target system installation from which you want to reconcile user records. For example: MYDB</td>
</tr>
<tr>
<td><strong>Object Type</strong></td>
<td>Enter the type of object whose values must be synchronized. For example: User</td>
</tr>
<tr>
<td><strong>Resource Object Name</strong></td>
<td>Enter the name of the resource object that is used for reconciliation. For example: MYDB User</td>
</tr>
</tbody>
</table>

**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. See the "Configuring Requests" chapter in *Oracle Fusion Middleware Developer’s Guide for Oracle Identity Manager* for complete information on configuring the request workflow.
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.

While running the testing utility, you must ensure that the connector is deployed and Oracle Identity Manager is running.

For detailed information about the testing utility, see Chapter 9, "Testing the Connector."

See Also:
- Section 4.6.4, "Configuring Request-Based Provisioning for Oracle Database" for a similar procedure for the Oracle Database
- Oracle Fusion Middleware User’s Guide for Identity Manager for information of managing requests
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:

- Section 9.1, "Running the Test Utility for the Oracle Database"
- Section 9.2, "Running the Test Utility for the MSSQL Database"
- Section 9.3, "Running the Test Utility for MySQL"
- Section 9.4, "Running the Test Utility for DB2"
- Section 9.5, "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.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Sample or Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Enter the type of operation that you want to test:</td>
<td>CONNECT</td>
</tr>
<tr>
<td></td>
<td>- CONNECT: To connect to the target system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- CREATE: To create a user in the target system, that is, provisioning</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- DELETE: To delete the user from the target system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- ENABLE: To enable the user in the target system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- DISABLE: To disable the user in the target system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- ADDROLE: To add/grant role to the user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- DELETEROLE: To delete/revoke role of the user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- ADDPRIVILEGE: To grant privilege to the user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- DELETEPRIVILEGE: To delete privilege to the user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- UPDATEPASSWORD: To update the password</td>
<td></td>
</tr>
<tr>
<td>ObjectType</td>
<td>Oracle supports only one user type</td>
<td>USER</td>
</tr>
<tr>
<td>Log Level</td>
<td>Enter one of the following values:</td>
<td>FINE</td>
</tr>
<tr>
<td></td>
<td>- OFF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- INFO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- FINE</td>
<td></td>
</tr>
<tr>
<td>OIM Login Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Auth Login Config File</td>
<td>Security Auth Login Config File</td>
<td>OIM_HOME/server/config/auth/wl.conf</td>
</tr>
<tr>
<td>OIM Connection URL</td>
<td>Enter the Oracle Identity Manager connection URL.</td>
<td>t3://OIM_HOST:PORT</td>
</tr>
<tr>
<td>Context Factory</td>
<td>Enter the name of the context factory.</td>
<td>weblogic.jndi.WLInitialContextFactory</td>
</tr>
<tr>
<td>OIM Admin User</td>
<td>Enter the name of the Oracle Identity Manager user.</td>
<td>xelsysadm</td>
</tr>
<tr>
<td>Database IT Resource Name</td>
<td>Enter the name of the IT resource from which connectivity information must be read.</td>
<td>Oracle DB</td>
</tr>
<tr>
<td>IT Resource Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process Form Fields and Query Code Keys</td>
<td>Enter the ID of the user to be provisioned/created.</td>
<td>TestUser</td>
</tr>
<tr>
<td>UserID</td>
<td>Note: UserID to be deleted or updated or enable/disable should be already existing on the target.</td>
<td></td>
</tr>
<tr>
<td><strong>NAME</strong></td>
<td>Enter the name of the user to be provisioned. This attribute will be used only while provisioning.</td>
<td>TestUser</td>
</tr>
<tr>
<td><strong>PASSWORD</strong></td>
<td>Enter the password of the user to be provisioned.</td>
<td>mypassw0r1</td>
</tr>
</tbody>
</table>
Running the Test Utility for the MSSQL Database

5. 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`

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.

### 9.2 Running the Test Utility for the MSSQL Database

Perform the following steps to run the test utility for MSSQL:

1. Drop the third party jar (sqljdbc4.jar) in the following directory:

   `OIM_HOME/DBUM-11.1.1.6.0/test/thirdparty`

---

**Table:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Sample or Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>authType</td>
<td>Enter one of the following authentication types:</td>
<td>PASSWORD</td>
</tr>
<tr>
<td></td>
<td>- PASSWORD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- GLOBAL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- EXTERNAL</td>
<td></td>
</tr>
<tr>
<td>globalDN</td>
<td>Enter the Global DN of the user. Provide the value for this attribute only when the authentication type is GLOBAL.</td>
<td>Global DN</td>
</tr>
<tr>
<td>tablespace</td>
<td>Default Tablespace</td>
<td></td>
</tr>
<tr>
<td>defaultQuota</td>
<td>Default Tablespace Quota (in MB).</td>
<td>defaultQuota</td>
</tr>
<tr>
<td>profile</td>
<td>Profile Name</td>
<td></td>
</tr>
<tr>
<td>tempTableSpace</td>
<td>Temporary Tablespace</td>
<td>tempTableSpace</td>
</tr>
</tbody>
</table>

**Roles to be added/deleted**

| Name            | Enter the role name.                                                        | CONNECT                |
|-----------------|                                                                            |                         |
| roles~DBRole~__  | Enter the role with admin option.                                          | WITH ADMIN OPTION      |
| NAME__           |                                                                            |                         |

**Privilege to be added/deleted**

| Name            | Enter the privilege name.                                                  | ALTER ANY ASSEMBLY     |
|-----------------|                                                                            |                         |
| privileges~DBPrivilege~__  | Enter the privilege with admin option.                                     | WITH ADMIN OPTION      |
| NAME__           |                                                                            |                         |
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 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

4. Update OIM_HOME/DBUM-11.1.1.6.0/test/config/mssqlconfig.properties with the required information.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Sample or Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Enter the type of operation that you want to test:</td>
<td>CONNECT</td>
</tr>
<tr>
<td></td>
<td>• CONNECT: To connect to the target system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• CREATE: To create a user in the target system, that is, provisioning</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DELETE: To delete the user from the target system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• ENABLE: To enable the user in the target system. This operation is supported</td>
<td></td>
</tr>
<tr>
<td></td>
<td>for the object type UserLogin only.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DISABLE: To disable the user in the target system. This operation is</td>
<td></td>
</tr>
<tr>
<td></td>
<td>supported for the object type UserLogin only.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• ADDROLE: To add/grant role to the user. This operation is supported for</td>
<td></td>
</tr>
<tr>
<td></td>
<td>object type User.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DELETEROLE: To delete/revoke role of the user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• UPDATEPASSWORD: To update the password. This operation is supported only</td>
<td></td>
</tr>
<tr>
<td></td>
<td>for the object type UserLogin.</td>
<td></td>
</tr>
<tr>
<td>ObjectType</td>
<td>MSSQL supports two user types: USER and USERLOGIN</td>
<td>USERLOGIN</td>
</tr>
<tr>
<td>Log Level</td>
<td>Enter one of the following values:</td>
<td>FINE</td>
</tr>
<tr>
<td></td>
<td>• OFF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• INFO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• FINE</td>
<td></td>
</tr>
<tr>
<td>OIM Login Data</td>
<td>Security Auth Login Config File</td>
<td>OIM_HOME/server/config/auth/wl.conf</td>
</tr>
<tr>
<td>OIM Connection URL</td>
<td>Enter the Oracle Identity Manager connection URL.</td>
<td>t3://OIM_HOST:PORT</td>
</tr>
<tr>
<td>Context Factory</td>
<td>Enter the name of the context factory.</td>
<td>weblogic:jndi.WLInitialContextFactory</td>
</tr>
<tr>
<td>OIM Admin User</td>
<td>Enter the name of the Oracle Identity Manager user.</td>
<td>xelsysadm</td>
</tr>
</tbody>
</table>
Running the Test Utility for MySQL

Testing the Connector 9-5

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.

9.3 Running the Test Utility for MySQL

Perform the following steps to run the test utility for MySQL:

1. 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Sample or Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database IT Resource Name</td>
<td>Enter the name of the IT resource from which connectivity information must be read.</td>
<td>MSSQL DB</td>
</tr>
<tr>
<td>IT Resource Name</td>
<td>Enter the name of the IT resource from which connectivity information must be read.</td>
<td>MSSQL DB</td>
</tr>
</tbody>
</table>

### Process Form Fields and Query Code Keys

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Sample or Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserID</td>
<td>Enter the ID of the user to be provisioned/created.</td>
<td>TestUser</td>
</tr>
<tr>
<td><strong>NAME</strong></td>
<td>Enter the name of the user to be provisioned. This attribute will be used only while provisioning.</td>
<td>TestUser</td>
</tr>
<tr>
<td><strong>PASSWORD</strong></td>
<td>Enter the password of the user to be provisioned.</td>
<td>mypassw0r1</td>
</tr>
<tr>
<td>loginName</td>
<td>Enter the login name of the user to be provisioned. This attribute will be used only while provisioning.</td>
<td>TestUser</td>
</tr>
<tr>
<td>authType</td>
<td>Enter one of the following authentication types:</td>
<td>SQL_SERVER_AUTHENTICATION</td>
</tr>
<tr>
<td></td>
<td>■ WINDOWS_AUTHENTICATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ SQL_SERVER_AUTHENTICATION</td>
<td></td>
</tr>
<tr>
<td>defaultDatabase</td>
<td>Default database, this is an optional attribute.</td>
<td>defaultDatabase</td>
</tr>
<tr>
<td>defaultLanguage</td>
<td>Default language, this is an optional attribute.</td>
<td>defaultLanguage</td>
</tr>
</tbody>
</table>

**Roles to be added/deleted**

| roles~DBRole~__NAME__ | Enter the role name for the user. | Role |

---

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.

9.3 Running the Test Utility for MySQL

Perform the following steps to run the test utility for MySQL:

1. 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
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 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
   ```

4. Update OIM_HOME/DBUM-11.1.1.6.0/test/config/mysqlconfig.properties with the required information.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other Data</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Enter the type of operation that you want to test:</td>
</tr>
<tr>
<td></td>
<td>CONNECT: To connect to the target system.</td>
</tr>
<tr>
<td></td>
<td>CREATE: To create a user in the target system, that is, provisioning.</td>
</tr>
<tr>
<td></td>
<td>DELETE: To delete the user from the target system.</td>
</tr>
<tr>
<td></td>
<td>UPDATEPASSWORD: To update the password.</td>
</tr>
<tr>
<td>ObjectType</td>
<td>MySQL supports only one user type. Do not modify this entry.</td>
</tr>
<tr>
<td>Log Level</td>
<td>Enter one of the following values:</td>
</tr>
<tr>
<td></td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>INFO</td>
</tr>
<tr>
<td></td>
<td>FINE</td>
</tr>
<tr>
<td>OIM Login Data</td>
<td></td>
</tr>
<tr>
<td>Security Auth Login Config File</td>
<td>Security Auth Login Config File</td>
</tr>
<tr>
<td>OIM Connection URL</td>
<td>Enter the Oracle Identity Manager connection URL. t3://OIM_HOST:PORT</td>
</tr>
<tr>
<td>Context Factory</td>
<td>Enter the name of the context factory. weblogic.jndi.WLInitialContextFactory</td>
</tr>
<tr>
<td>OIM Admin User</td>
<td>Enter the name of the Oracle Identity Manager user. xelsysadm</td>
</tr>
<tr>
<td>Database IT Resource Name</td>
<td></td>
</tr>
<tr>
<td>IT Resource Name</td>
<td>Enter the name of the IT resource from which connectivity information must be read. MySQL DB</td>
</tr>
<tr>
<td>Process Form Fields and Query Code Keys</td>
<td></td>
</tr>
<tr>
<td>UserID</td>
<td>Enter the ID of the user to be provisioned/created.</td>
</tr>
<tr>
<td>Note: UserID to be deleted or updated should be already existing on the target.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TestUser</td>
</tr>
</tbody>
</table>

---

9-6 Oracle Identity Manager Connector Guide for Database User Management
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.

3. 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
java -cp '..lib/*:..thirdparty/*'
oracle.iam.connectors.dbum.testutil.prov.ProvisioningUtility
..config/db2config.properties
..bundle/org.identityconnectors.dbum-1.0.1116.jar
```

4. Update OIM_HOME/DBUM-11.1.1.6.0/test/config/db2config.properties with the required information.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Sample or Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NAME</strong></td>
<td>Enter the name of the user to be provisioned. This attribute will be used only while provisioning.</td>
<td>TestUser</td>
</tr>
<tr>
<td><strong>PASSWORD</strong></td>
<td>Enter the password of the user to be provisioned.</td>
<td>mypassword</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Sample or Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other Data</td>
<td>Enter the type of operation that you want to test:</td>
<td>CONNECT</td>
</tr>
<tr>
<td>Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONNECT</td>
<td>To connect to the target system.</td>
<td></td>
</tr>
<tr>
<td>CREATE</td>
<td>To create a user in the target system, that is, provisioning.</td>
<td></td>
</tr>
<tr>
<td>DELETE</td>
<td>To delete the user from the target system.</td>
<td></td>
</tr>
<tr>
<td>ENABLE</td>
<td>To enable the user in the target system.</td>
<td></td>
</tr>
<tr>
<td>DISABLE</td>
<td>To disable the user in the target system.</td>
<td></td>
</tr>
</tbody>
</table>
Running the Test Utility for Sybase

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.

9.5 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Sample or Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectType</td>
<td>DB2 supports two user types: USER and GROUP</td>
<td>USER</td>
</tr>
<tr>
<td>Log Level</td>
<td>Enter one of the following values:</td>
<td>FINE</td>
</tr>
<tr>
<td></td>
<td>OFF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INFO</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FINE</td>
<td></td>
</tr>
</tbody>
</table>

### OIM Login Data

<table>
<thead>
<tr>
<th>Security Auth Login Config File</th>
<th>Security Auth Login Config File</th>
<th>$OIM_HOME/server/config/authwl.conf</th>
</tr>
</thead>
<tbody>
<tr>
<td>OIM Connection URL</td>
<td>Enter the Oracle Identity Manager connection URL.</td>
<td>t3://OIM_HOST:PORT</td>
</tr>
<tr>
<td>Context Factory</td>
<td>Enter the name of the context factory.</td>
<td>weblogic:jndi.WLInitialContextFactory</td>
</tr>
<tr>
<td>OIM Admin User</td>
<td>Enter the name of the Oracle Identity Manager user.</td>
<td>xelsysadm</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>UserID</th>
<th>Enter the ID of the user to be provisioned/created.</th>
<th>Dummy_Account</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Note: UserID to be deleted or updated or enable/disable should be already existing on the target.</td>
<td></td>
</tr>
<tr>
<td><strong>NAME</strong></td>
<td>Enter the name of the user to be provisioned. This attribute will be used only while provisioning.</td>
<td>Dummy_Account</td>
</tr>
<tr>
<td>userType</td>
<td>Enter the type of the user to be provisioned.</td>
<td>USER</td>
</tr>
</tbody>
</table>

Note:

- UserID to be deleted or updated or enable/disable should be already existing on the target.
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
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.

### Other Data

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Sample or Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Enter the type of operation that you want to test: 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.</td>
<td>CONNECT</td>
</tr>
<tr>
<td>Log Level</td>
<td>Enter one of the following values:</td>
<td>FINE</td>
</tr>
<tr>
<td>ObjectType</td>
<td>Sybase supports two user types: USER and USERLOGIN USERLOGIN</td>
<td>USERLOGIN</td>
</tr>
<tr>
<td>OIM Login Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Auth Login Config File</td>
<td>Security Auth Login Config File</td>
<td>$OIM_HOME/server/config/authwl.conf</td>
</tr>
<tr>
<td>OIM Connection URL</td>
<td>Enter the Oracle Identity Manager connection URL.</td>
<td>t3://OIM_HOST:PORT</td>
</tr>
<tr>
<td>Context Factory</td>
<td>Enter the name of the context factory.</td>
<td>weblogic.jndi.WLInitialContextFactory</td>
</tr>
<tr>
<td>OIM Admin User</td>
<td>Enter the name of the Oracle Identity Manager user.</td>
<td>xelsysadm</td>
</tr>
</tbody>
</table>
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.
The following sections describe known issues and workarounds associated with this release of the connector:

- Section 10.1, "Connector Issues"
- Section 10.2, "Oracle Identity Manager Issues"
- Section 10.3, "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:

- Section 10.2.1, "Update of a Child Table does not Work as Expected"
- Section 10.2.2, "User Remains in Provisioned Status After a Trusted Delete Reconciliation Run"
- Section 10.2.3, "Target Reconciliation Does Not Revoke Roles and Privileges"
- Section 10.2.4, "The With Grant Option in Child Data Fails to Appear in Entitlement List"
- Section 10.2.5, "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:

- Section 10.3.1, "Creation of a JDBC Connection Results in an Indefinite Wait Time"
- Section 10.3.2, "Incremental Reconciliation Cannot Process Role Updateds in MSSQL"
- Section 10.3.3, "Relevant Error Log not Provided While Provisioning a Duplicate User"
- Section 10.3.4, "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 Updateds 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.
Index

A
action scripts, 3-51, 4-54, 5-44, 6-45, 7-44
architecture, 1-4
authenticated users
  password, 1-8, 1-12

B
bat scripts
  CMD, 3-51, 4-55, 5-45, 6-46, 7-44
ConnectorServer.bat, 2-8
batched reconciliation, 1-9

C
certified components, 1-2
changing input locale, 2-10
clearing server cache, 2-11
CMD batch script, 3-51, 4-55, 5-45, 6-46, 7-44
cols, 2-3
components, certified, 1-2
configuring
  action scripts, 3-51, 4-54, 5-44, 6-45, 7-44
direct provisioning, 3-26, 4-29, 5-21, 6-22, 7-20
request-based provisioning, 3-27, 4-30, 5-22, 6-23, 7-21
scheduled tasks, 3-15, 4-17, 5-11, 6-12, 7-13
target system as trusted source, 3-19, 4-21, 5-15, 6-15
configuring connector
database vault realms, 2-12
configuring SSL, 1-11
configuring transformation, 1-10
configuring validation, 1-10
connection pooling, 2-12
connector
  upgrading, 2-27
connector architecture, 1-4
connector features, 1-6
connector files and directories
description, 2-1
installation media file, 2-1
Connector Installer, 2-3
connector installer, 2-3
connector server, 2-8
installing, 2-6
installing and configuring, 2-6
running, 2-8
uninstalling, 2-8
ConnectorServer.bat, 2-8
ConnectorServer.properties file, 2-7
creating users
  externally-authenticated, 4-27
globally-authenticated, 4-26
password-authenticated, 4-26

D
data encryption and integrity, 3-1, 4-1, 5-1, 6-1, 7-1
DB2, 6-1
defining
  IT resources, 2-20
deployment manager
  importing request datasets, 3-28, 4-31, 5-23, 6-23, 7-22

E
enabling logging, 2-13, 2-26
exclusion lists, 3-49, 4-53, 5-43, 6-44
external, 1-12
eexternally-authenticated users, 1-8, 1-12

F
features of connector, 1-6
Files and Directories, 2-1
files and directories of the connector
  See connector files and directories
Filter parameter, 3-22, 4-24, 5-18, 6-18
filtered reconciliation, 4-24, 5-17, 6-18
  See limited reconciliation, 1-8
filtering
  target system records, 3-22, 4-24, 5-18, 6-18
full reconciliation, 1-8

G
global, 1-12
globalization features, 1-3
globally-authenticated users, 1-8, 1-12
Identity Connector Framework, 3-51, 4-55, 5-44, 6-45, 7-44
importing request datasets
  deployment manager, 3-28, 4-31, 5-23, 6-23, 7-22
incremental reconciliation, 1-8
input locale changing, 2-10
installation, 2-3
  preinstallation, 2-1
installing
  connector server, 2-6
installing connector, 2-1, 2-3, 2-9
IT resource
  configuring for Connector Server, 2-24
IT resources
  defining, 2-20
  parameters, 2-20
Java Connector Server
  installing, 2-6
  uninstalling, 2-8
Java connector server
  running, 2-8
JDBC URL and connection properties, 4-3, 5-2, 6-2, 7-2
  MSSQL, 3-2
limited reconciliation, 1-8, 4-24, 5-17, 6-18
logging enabling, 2-13, 2-26
lookup definitions
  DB2, 6-3
  Lookup.DBUM.SQLServer.Configuration, 6-21, 6-37
  Lookup.DBUM.MSSQL.Configuration, 2-21, 3-6, 3-25, 3-42
  Lookup.DBUM.MSSQL.TrustedRecon.Configuration, 2-21
  Lookup.DBUM.MySQL.Configuration, 5-20, 5-36
  Lookup.DBUM.Oracle.Configuration, 2-21, 4-28, 4-46
  Lookup.DBUM.Oracle.TrustedRecon.Configuration, 2-21
  Lookup.DBUM.Sysbase.Configuration, 7-19, 7-35
  MSSQL, 3-4
  MySQL, 5-3
  Oracle Database, 4-7
  Sybase, 7-3
lookup definitions for attribute mappings
  DB2, 6-6
  MSSQL Login, 3-8
  MSSQL User, 3-9
  MySQL, 5-6
  Oracle Database, 4-10
  Sybase Login, 7-7
  Sybase User, 7-8
lookup definitions for configurations
  DB2, 6-5
MSSQL, 3-6
MySQL, 5-5
Oracle Database, 4-8
Sybase, 7-4
lookup definitions for data transformation
  DB2, 6-9
  MSSQL, 3-12
  MySQL, 5-9
  Oracle Database, 4-13
  Sybase, 7-10
lookup definitions for data validation
  DB2, 6-9
  MSSQL, 3-12
  MySQL, 5-9
  Oracle Database, 4-14
  Sybase, 7-9
lookup definitions for exclusion lists
  DB2, 6-8
  MSSQL, 3-11
  MySQL, 5-8
  Oracle Database, 4-13
  Sybase, 7-10
lookup field synchronization, 3-17, 4-18, 6-13
  DB2, 6-3
  MSSQL, 3-4
  MySQL, 5-4
  Oracle Database, 4-7
  scheduled tasks, 3-12, 4-14, 5-9, 6-10, 7-11
  Sybase, 7-3
lookup fields, 3-17, 4-18, 6-13
Microsoft SQL Server, 3-1
multilanguage support, 1-3
multiple installations, 3-42
multiple trusted source reconciliation, 4-49, 5-39, 6-40
multiple versions of the target system
  configuring, 2-22
MySQL, 5-1
Oracle Database, 4-1
Oracle Database Vault, 1-12, 2-12, 2-22
Oracle Database Vault realms, 1-12
parameters of IT resources, 2-20
postcloning, 2-30
provisioning
  configuring action scripts, 3-51, 4-54, 5-44, 6-45, 7-44
  DB2, 6-19
direct, 3-26, 4-29, 5-21, 6-22, 7-20
direct provisioning, 3-26, 4-29, 5-21, 6-22, 7-20
direct to request-based, 3-30, 4-33, 5-25, 6-26, 7-24
guidelines for DB2, 6-19
guidelines for MSSQL, 3-23
guidelines for MySQL, 5-19
guidelines for Oracle Database, 4-26
guidelines for Sybase, 7-17
MSSQL, 3-23
MySQL, 5-18
Oracle Database, 4-25
process, 4-27, 5-19, 6-20, 7-18
provisioning triggered by policy changes, 3-24, 4-27, 5-19, 6-20, 7-18
queries, 3-33, 4-36, 5-27, 6-28, 7-26
request-based, 3-27, 4-30, 5-22, 6-23, 7-21
request-based provisioning, 3-24, 4-27, 5-19, 6-20, 7-18
request-based to direct, 3-30, 4-33, 5-25, 6-25, 7-24
Sybase, 7-17

Q
queries
configuring guidelines, 4-35, 5-27, 6-27, 7-26
list of values, 3-37, 4-40, 5-30, 6-32, 7-30
patching bundle, 8-4
provisioning, 3-33, 4-36, 5-27, 6-28, 7-26
reconciliation, 3-35, 4-38, 5-29, 6-30, 7-28
search, 3-35, 4-38, 5-29, 6-30, 7-28
updating, 3-38, 4-41, 5-31, 6-32, 7-30

R
Reconciliation, 4-19, 5-13, 6-14, 7-15
reconciliation
batched, 1-9, 4-25, 5-18, 6-18
DB2, 6-13
full, 1-8, 3-22, 4-23, 5-17, 6-18, 7-17
incremental, 1-8, 3-22, 4-25
limited, 3-22, 4-24, 5-17, 6-18
MSSQL, 3-16
MySQL, 5-13
Oracle Database, 4-18
queries, 3-35, 4-38, 5-29, 6-30, 7-28
queries for MSSQL, 3-18
queries for Oracle Database, 4-20
Sybase, 7-14
reconciliation action rule
target resource reconciliation, 4-23, 6-17, 7-16
trusted source reconciliation, 4-23, 5-16, 6-17, 7-16
reconciliation rule
target resource reconciliation, 4-22, 5-16, 6-16, 7-16
trusted source reconciliation, 4-22, 6-16, 7-16

S
scheduled tasks
attributes, 3-13, 4-15, 5-10, 6-11, 7-12
configuring for DB2, 6-12
configuring for MSSQL, 3-15
configuring for MySQL, 5-11
configuring for Oracle Database, 4-17
configuring for Sybase, 7-13
lookup field synchronization, 3-12, 4-14, 5-9, 6-10,