

Oracle® Business Intelligence Applications

Installing Business Intelligence Applications



12c (11.1.1.10.3 PS4)

F83292-03

December 2023

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

ORACLE®

Oracle Business Intelligence Applications Installing Business Intelligence Applications, 12c (11.1.1.10.3 PS4)

F83292-03

Copyright © 2013, 2023, Oracle and/or its affiliates.

Primary Author: Hemala Vivek

Contributors: Nick Fry, Christine Jacobs, Padma Rao

Contributors: Oracle Business Intelligence development, product management, and quality assurance teams.

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

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

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

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

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

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

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

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

Contents

Preface

Functional Setup Manager - Statement of Direction	vii
Audience	vii
Documentation Accessibility	vii
Diversity and Inclusion	vii
Related Documents	viii
Conventions	viii

1 Oracle BI Applications Architecture and Installation Overview

Oracle BI Applications Architecture	1-1
Oracle BI Applications Installation Roadmap	1-2

2 Preinstallation and Deployment Requirements for Oracle BI Applications

General Guidelines for Setting Up Databases for Oracle BI Applications	2-1
Guidelines for Setting Up Oracle Business Analytics Warehouse Databases	2-1
Why Use a Separate Database for the Oracle Business Analytics Warehouse?	2-1
Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse	2-2
General Guidelines for Oracle Databases	2-2
Guidelines for Using Oracle Template Files	2-3

3 Installing and Setting Up Oracle BI Applications

Hardware Requirements	3-1
Installation Prerequisites for Oracle BI Applications	3-1
Install and Configure Oracle BI Applications	3-1
Install JDK	3-2
Install Oracle Fusion Middleware Infrastructure (WebLogic Server)	3-2
Install Oracle Analytics Server	3-3
Create Schemas Using RCU	3-4
Install Oracle Data Integrator	3-10
Install Oracle BI Applications	3-10

Create Schemas for Oracle BI Applications	3-11
Create BIApps Schema on Oracle Autonomous Transaction Processing (ATP)	3-18
Set the BIAPPS Schema User Password as Unlimited	3-20
Upgrade OPatch	3-21
Apply Mandatory Patches	3-21
Configure Oracle Analytics Server	3-22
Edit Your Input File	3-23
Configure Oracle BI Applications	3-26
Configure Oracle BI Applications with Oracle Autonomous Data Warehouse	3-26
Load Translation Data	3-28
Load Translation Data on Oracle Autonomous Transaction Processing (ATP)	3-29
Configure Oracle Data Integrator Studio	3-31
Configuring Authentication	3-31
Setting the JPS Config File Location	3-31
Starting Oracle Data Integrator Studio	3-31
Connecting to the Master Repository	3-31
Regenerating the Security Files	3-34
Use a RAC Database	3-34
Use an External LDAP	3-35
Configure Oracle BI Applications for High Availability	3-35
Configure High Availability with Automated Scripts	3-36
Configuring High Availability with Manual Steps	3-37
Configuring High Availability for ODI	3-39
Configuring Oracle HTTP Server for High Availability	3-40
Performing Post-Configure Steps for High Availability	3-41
Creating BI Schemas Using RCU with a RAC Database	3-43
Configure SSL	3-43
Uninstalling Oracle BI Applications	3-43

4 Performing Post-Installation System Setup Tasks

Setting the Business Analytics Warehouse Connection in Oracle Data Integrator	4-1
Performing Setup Steps for On-Premises Oracle Fusion Applications Sources	4-2
Creating a User for ETL	4-2
Configuring the Oracle BI Repository for ETL	4-2
Registering Source Systems and Propagating Connection Details to Oracle Data Integrator	4-6
Enabling Offerings for Deployment	4-9
Disabling a Source Instance	4-10
Setting Languages for Data Load into the Business Analytics Warehouse	4-10
Running the Domains Load Plan	4-11
Mapping Externally Conformed Domains	4-12

User Access to Configuration Manager, FSM, and Oracle Data Integrator	4-12
Editing Preferred Currency Display Names and Enabling Document Currency	4-13
Enabling Document Currency	4-13
Enabling Initialization Blocks	4-15
Trimming the Repository	4-16
Extracting Projects From Full Repository	4-16
Merging Repositories	4-17
Next Steps After Setup — Functional Configuration	4-18

5 Setting Up Fusion Applications Cloud Data Sources

Overview of Oracle Fusion Applications Cloud Source System Support	5-1
Fusion Applications Cloud Source System Deployment Options	5-2
Fusion Application Cloud Source System Deployment Roadmap	5-3
Creating Service Requests for Oracle Business Intelligence Cloud Connector Console	5-4
Provisioning a User for Export and Import of BI Data from Fusion UCM	5-5
Provisioning a User for Oracle BI Cloud Connector Console Access	5-5
Registering the Fusion Applications Source	5-6
Creating a Load Plan and Specifying Offerings and Fact Groups to Load	5-9
Extracting Data into Universal Content Management using Oracle BI Cloud Connector Console	5-11
Synchronizing Deletes for a Cloud Extract	5-14
Configuring Proxy Settings for WebLogic Server	5-15
Managing Fusion Flex Extensions in Release 11.1.1.10.3	5-16
Setting Up Key Flex Fields for Financials Fusion Applications Source Data	5-21
Using Oracle BI Cloud Connector Console to Configure Cloud Extract	5-21
Using ODI to Set Up Key Flex Fields	5-24
Setting Up Key Flex Fields for HCM Fusion Applications Source Data	5-29

6 Setting Up Oracle Taleo Cloud Data Source

Registering a Taleo Cloud Source System	6-1
---	-----

7 Deploying the ODI Repository for Non-Oracle Source Databases

Exporting Security Settings from Original ODI Repository	7-2
Exporting Topology Settings from Original ODI Repository	7-4
Creating a New Master Repository	7-7
Connecting to the New Master Repository	7-13
Creating a New Work Repository	7-14
Importing Source Adaptor Metadata Content into the New ODI Repository	7-18
Preparing to Import Source Adaptor Metadata Content	7-18

Running the Command Line Utility to Import Source Adaptor Metadata Content	7-21
Importing Security Settings in to the ODI Repository	7-32
Importing Connection Details that Configure the Data Servers	7-32

A Generating DDL and Assigning Tablespaces to Tables and Indexes

Overview of Oracle Business Analytics Warehouse Tables	A-1
Generating the Oracle Business Analytics Warehouse DDL	A-2
Oracle BI Applications Patches	A-4
Assigning Tablespaces	A-4

B Integrating Dashboards and Siebel Applications

Overview of Embedding Oracle Analytics in Siebel Application	B-1
Configure Oracle HTTP Server	B-2
Update the Siebel Application to Find Oracle Analytics Server Through HTTP Server	B-3
About Configuring Action Links	B-3
Access Optional Analytics Applications	B-3
Customize Oracle BI Applications Home Page and Dashboards	B-4
Configure a New Home Page to Use a Symbolic URL	B-4
Determine the Oracle Analytics Report Path Argument	B-5
Add Views for Custom Dashboards	B-6
View Preconfigured Dashboards	B-6
Exposing Dashboards and Requests	B-6
Locating Dashboards, Page Names, and Reports	B-6

C Configuring Oracle Analytics Server Domain

Create BI Schema on Oracle ATP Database	C-1
Configure Oracle Analytics Server Domain Using Configuration Assistant	C-3
Configure Oracle Analytics Server Domain on Oracle ATP	C-6

Preface

Oracle Business Intelligence Applications (Oracle BI Applications) is a comprehensive suite of prebuilt solutions.

It delivers pervasive intelligence across an organization, empowering users at all levels — from front line operational users to senior management - with the key information they need to maximize effectiveness. Intuitive and role-based, these solutions transform and integrate data from a range of enterprise sources and corporate data warehouses into actionable insights that enables more effective actions, decisions, and processes.

Oracle BI Applications is built on Oracle Analytics Server, a comprehensive set of enterprise business intelligence tools and infrastructure, including a scalable and efficient query and analysis server, an ad hoc query and analysis tool, interactive dashboards, proactive intelligence and alerts, and an enterprise reporting engine.

Functional Setup Manager - Statement of Direction

Functional Setup Manager (FSM) has been desupported in the current release. Ignore the references to FSM that you might see in the online Help.

Audience

This document is intended for managers and implementers of Oracle BI Applications.

Documentation Accessibility

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

Access to Oracle Support

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

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry

standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Documents

See the Oracle BI Applications documentation library for the complete set of Oracle BI Applications documents.

Conventions

These text conventions are used in this document.

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

1

Oracle BI Applications Architecture and Installation Overview

This section provides an overview of the Oracle Business Intelligence Applications (Oracle BI Applications) architecture and installation process.

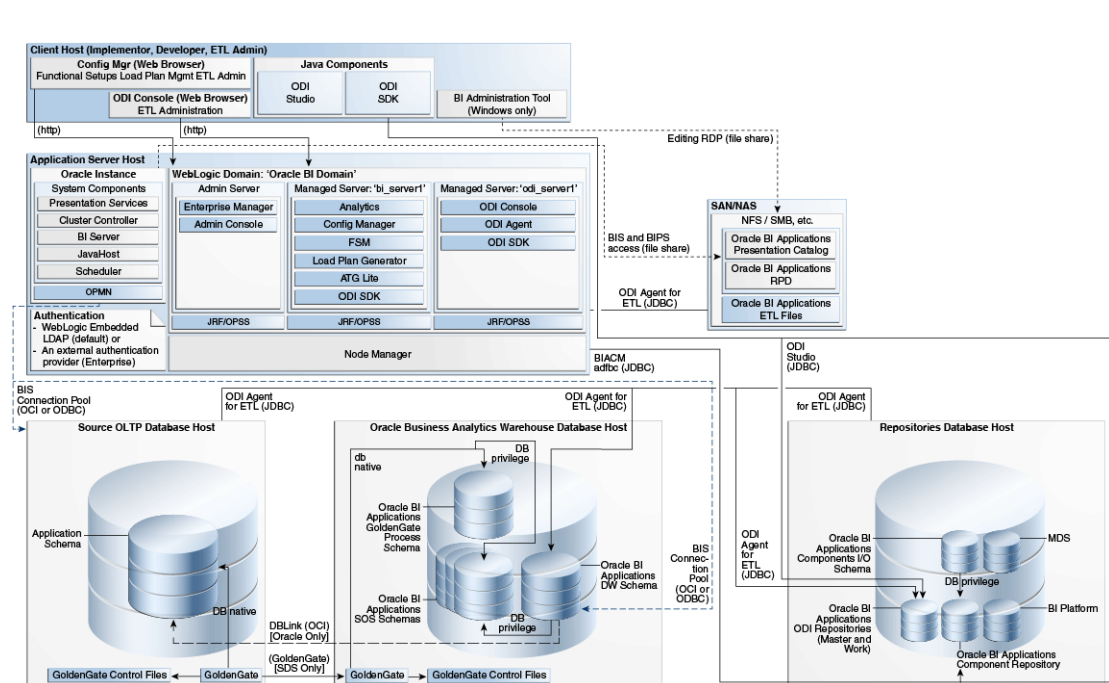
Topics:

- [Oracle BI Applications Architecture](#)
- [Oracle BI Applications Installation Roadmap](#)

Oracle BI Applications Architecture

Review the Oracle BI Applications deployment architecture, which is comprised of several components and repositories.

Oracle BI Applications Deployment Architecture Diagram



Components Deployed Into Manager Server "bi_server1"

- Oracle Business Intelligence Applications Configuration Manager — This component is a web application that enables you to perform system setups, functionally configure your Oracle BI Applications, and define, generate, run, and monitor load plans.
- Load Plan Generator — This component is a set of jar files used to create load plans in the ODI Repository.

Components Deployed Into Managed Server "odi_server1"

- ODI Console — This component enables you to control and monitor ETL.
- ODI Agent — This component is a Java EE Agent, which handles schedules and orchestrates sessions.

Oracle BI Applications Repositories

- Oracle Business Analytics Warehouse — The Oracle Business Analytics Warehouse is a unified data repository for all customer-centric data, which supports the analytical requirements of the supported source systems. The Oracle Business Analytics Warehouse is supported only on Oracle Database.
- Oracle Business Intelligence Applications Components Repository (BIACOMP) — This is the repository for Configuration Manager. It contains load plan definitions as well as Oracle BI Applications product hierarchy, setup objects (such as parameters and domain mappings), and a list of functional tasks. This repository is supported only on Oracle Database.
- ODI Repository for Oracle BI Applications — This repository contains the Oracle BI Applications-specific prebuilt ETL logic.

Client Tier

As depicted in the figure, the client host accesses the Configuration Manager and ODI Console through web browsers. Install ODI Studio on a developer client system. Install ODI SDK along with ODI Studio to work with Oracle BI Applications.

Additional Tools

Oracle GoldenGate is an optional tool integrated with Oracle BI Applications. You use this tool to optimize ETL loads and reduce impact on the source system, using a source-dependent data store.

To deploy Oracle GoldenGate in an Oracle BI Applications environment, see Setup Step: Install Oracle GoldenGate on Source and Target Systems in the *Oracle Business Intelligence Applications Administrator's Guide*.

ETL Architecture

To learn about ETL architecture, see About ETL Architecture in the *Oracle Business Intelligence Applications ETL Guide*.

Oracle BI Applications Installation Roadmap

Review the high-level roadmap for installing and configuring Oracle BI Applications.

1. Review the [Oracle® Fusion Middleware System Requirements and Specifications 12c \(12.2.1.4.0\)](#).
2. Review preinstallation and deployment requirements.
3. Install these prerequisites:
 - Oracle Fusion Middleware Infrastructure 12.2.1.4.0
 - Oracle Analytics Server

Refer to the Oracle BI Applications certification matrix on Oracle Technology Network for the supported version of Oracle Analytics Server.

- Oracle Data Integrator 12.2.1.4
- 4. Create Oracle Analytics Server schemas using Oracle Fusion Middleware Repository Creation Utility (RCU).
- 5. Install Oracle BI Applications.

In this step, you run the Business Analytics Applications Suite installer to install the binary files for Oracle BI Applications.

- a. [Create Schemas for Oracle BI Applications](#)
 - b. [Create BIApps Schema on Oracle Autonomous Transaction Processing \(ATP\)](#)
 - c. [Upgrade OPatch](#)
6. Apply Oracle Fusion Middleware Platform patches.
 7. Configure Oracle Analytics Server and create BI Domain.
 8. Configure Oracle BI Applications.

In this step, you configure Oracle BI Applications with the configuration script.

You perform system setups to complete integration of the Oracle BI Applications components, and to prepare the system for functional configurations and data loads.

9. Configure non-Oracle source databases.
This step is required only if your source database isn't Oracle Database.
10. If you're deploying Oracle Fusion Applications Cloud source system data, then follow the steps in [Setting Up Fusion Applications Cloud Data Sources](#).
11. If you're deploying Key Flex Fields for Financials, then before you execute this load plan, you must follow the steps in [Setting Up Key Flex Fields for Financials Fusion Applications Source Data](#).

2

Preinstallation and Deployment Requirements for Oracle BI Applications

This section describes preinstallation and deployment requirements for Oracle Business Intelligence Applications (Oracle BI Applications).

Topics:

- [General Guidelines for Setting Up Databases for Oracle BI Applications](#)
- [Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse](#)

General Guidelines for Setting Up Databases for Oracle BI Applications

These guidelines help you set up the data warehouse physical database and explain why a separate database is necessary for Oracle BI Applications.

- [Guidelines for Setting Up Oracle Business Analytics Warehouse Databases](#)
- [Why Use a Separate Database for the Oracle Business Analytics Warehouse?](#)

Guidelines for Setting Up Oracle Business Analytics Warehouse Databases

These guidelines help you set up the data warehouse physical database for performance and growth.

- Allocate approximately 50 to 70 percent of the total available server memory to the database, assuming no other application is running on the same server.
- At a minimum, separate the data and index tablespaces. Create more tablespaces to separate heavily-used tables and their indexes.
- Oracle recommends using 8 KB block size for Oracle warehouses.
- If you're using multiple disk storage systems, stripe the tablespace containers and files across as many disks as possible.
- Raw devices for tablespaces provide better performance as compared to cooked file systems.
- RAID-5 is known to give a good balance of performance and availability.

Why Use a Separate Database for the Oracle Business Analytics Warehouse?

To avoid performance issues, don't put Oracle Business Analytics Warehouse in the same database as the transactional database. The transactional database is structured as an

online transaction processing (OLTP) database, whereas the Oracle Business Analytics Warehouse is structured as an online analytical processing (OLAP) database, each optimized for its own purpose.

The reasons for not combining the two databases are:

- ETL is configured to maximize hardware resources, therefore the warehouse shouldn't share any resources with any other projects.
- The analytical queries interfere with normal use of the transactional database, which is entering and managing individual transactions.
- The data in a transactional database is normalized for efficient updates. Transactional queries join several normalized tables and will be slow (as opposed to pre-joined, de-normalized analytical tables).
- Historical data can't be purged from a transactional database, even if it's not required for current transaction processing, because you need it for analysis. (By contrast, the analytical database is the warehouse for historical as well as current data.) The transactional database slows down further.
- Transactional databases are tuned for one specific application, and it's not productive to use these separate transactional databases for analytical queries that usually span more than one functional application.
- The analytical database can be specifically tuned for the analytical queries and Extract-Transform-Load (ETL) processing. These are quite different from transactional database requirements.

Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse

These guidelines help you to optimize Oracle database performance and configure Oracle Business Analytics Warehouse on Oracle databases.

Topics:

- [General Guidelines for Oracle Databases](#)
- [Guidelines for Using Oracle Template Files](#)

General Guidelines for Oracle Databases

These additional suggestions help you to optimize performance for Oracle databases used for Oracle Business Analytics Warehouse.

- Oracle BI Applications on Oracle databases support only binary sorting. If you're running an Oracle client, do one of the following:
 - Set the NLS_SORT parameter to BINARY.
 - Choose a NLS_LANG setting that includes binary.

These settings are required for adequate performance from the dedicated web client.

- It's recommended that you gather workload system statistics.
- Set the number of log file groups to 4.

- To increase data throughput between Oracle BI Server and the Oracle database, configure the SDU and TDU settings as detailed below. The default is 8 KB and can be increased to 2 MB.
- On the server side, make the following changes:
 1. In the `sqlnet.ora` file, add applicable value for the **DEFAULT_SDU_SIZE** parameter. For example, `DEFAULT_SDU_SIZE = 65535`.
 2. In `listener.ora`, modify your listener entry by adding `SDU=`. For example,

```
<your listener name > =
(DESCRIPTION_LIST =
(DESCRIPTION = (SDU = 65535)
(ADDRESS = (PROTOCOL = TCP) (HOST = myhost) (PORT = 1521))
(ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1521)) ) )
```

3. Restart the listener.
- On the client side, make the following changes:
 1. In the `sqlnet.ora` file, add applicable value for the **DEFAULT_SDU_SIZE** parameter. For example, `DEFAULT_SDU_SIZE = 65535`.
 2. In `Tnsnames.ora`, modify the TNS alias by adding `SDU=` and `TDU=`. For example,

```
myhost_orcl.world=
(DESCRIPTION= (SDU = 65535 ) (TDU = 65535)
ADDRESS = (PROTOCOL = TCP) (HOST=myhost) (PORT=1521))
CONNECT_DATA=(SID=ORCL))
```

Guidelines for Using Oracle Template Files

To configure the Oracle Business Analytics Warehouse on Oracle databases more easily, refer to the parameter template file `init11gR2_FusionApplications_template.ora` file (this common template file applies to all sources in the Oracle BI Applications certified warehouse database) that is available in `/ORACLE_HOME/bi/biapps/etl`.

The parameter template file provides parameter guidelines based on the cost-based optimizer for the Oracle database version mentioned in the Certification Matrix. Use these guidelines as a starting point. Review the recommendations in the template file and make changes based on your specific database sizes, data shape, server size (CPU and memory), and the type of storage. Also, change the settings based on performance monitoring and tuning considerations. While making changes, ensure that you retain the following parameters:

- `db_block_size = 8192`
- `processes = 2000`
- `parallel_adaptive_multi_user = FALSE`

3

Installing and Setting Up Oracle BI Applications

Follow these instructions for installing and setting up Oracle Business Intelligence Applications (Oracle BI Applications).

Topics:

- [Hardware Requirements](#)
- [Installation Prerequisites for Oracle BI Applications](#)
- [Installing and Configuring Oracle BI Applications](#)
- [Uninstalling Oracle BI Applications](#)

Hardware Requirements

Regarding the hardware and performance requirements for Oracle BI Applications, refer to the Oracle BI Applications certification matrix on Oracle Technology Network.

For additional information about hardware requirements for Oracle Analytics Server and Oracle Data Integrator (ODI), see Oracle Fusion Middleware Supported System Configurations on Oracle Technology Network.

Installation Prerequisites for Oracle BI Applications

You must install certain components before you can install Oracle BI Applications.

Prior to installing an Oracle database, refer to the Oracle BI Applications certification matrix on Oracle Technology Network for the certified Oracle databases. While installing the database, ensure that you don't select a container database.

If you're planning to use Oracle Autonomous Transaction Processing (ATP), then you must also configure Oracle Autonomous Datawarehouse (ADW). On ATP, create the BI platform schema, ODI schema, and BI_COMP schema. Create rest of the schemas such as DW and SDS on ADW.

Install and Configure Oracle BI Applications

Use these instructions to install and configure Oracle BI Applications.

Oracle BI Applications requires:

- Oracle Fusion Middleware which includes the Oracle WebLogic Server 12c (12.2.1.4)
- Oracle Analytics Server
Refer to the Oracle BI Applications certification matrix on Oracle Technology Network for the supported version of Oracle Analytics Server.
- Oracle Data Integrator (12.2.1.4)

You must perform the following procedures in the order they are listed:

1. [Install JDK](#)
2. [Install Oracle Fusion Middleware Infrastructure \(WebLogic Server\)](#)
3. [Install Oracle Analytics Server](#)
4. [Create Schemas Using RCU](#)
5. [Install Oracle Data Integrator](#)
6. [Install Oracle BI Applications](#)
7. [Create Schemas for Oracle BI Applications](#)
8. [Apply Mandatory Patches](#)
9. [Configure Oracle Analytics Server](#)
10. [Edit Your Input File](#)
11. [Configure Oracle BI Applications](#)
12. [Load Translation Data](#)
13. [Configure Oracle Data Integrator Studio](#)
14. [Use a RAC Database](#)
15. [Use an External LDAP](#)
16. [Configure Oracle BI Applications for High Availability](#)
17. [Configure SSL](#)

Install JDK

To install the recommended JDK version, refer to the Oracle BI Applications certification matrix on Oracle Technology Network.

Refer to the JDK installation document for detailed instructions.

Note:

Ensure that you remove any other version of Java present in the system before installing JDK on your local machine and not in any NFS path. You may set `JAVA_HOME` as the environment variable for further use. You may set `JAVA_HOME` as the environment variable for further use. Add `$JAVA_HOME/bin` in your `PATH` variable. You may use `/usr/sbin/alternatives --config java` to configure the recommended version as the default version of Java.

Install Oracle Fusion Middleware Infrastructure (WebLogic Server)

Oracle BI Applications requires Oracle Fusion Middleware Infrastructure 12.2.1.4.

1. Download the Oracle Fusion Middleware Infrastructure Installer from the [Download page for Oracle Fusion Middleware Software](#).

2. For Linux systems, review the general UNIX operating system requirements, and verify the locale and open file limit settings as described in [Oracle® Fusion Middleware System Requirements and Specifications 12c \(12.2.1.4.0\)](#). Oracle recommends a high value for ulimit. A low value of open file limit and the number of processes limit can cause installation failure. As the root user, do the following:

- a. Edit the `/etc/security/limits.conf` file and save the following values.

```
* hard nofile 999999
* soft nofile 999999
* soft nproc 131072
* hard nproc 131072
* soft core unlimited
* hard core unlimited
```

The `nofile` values represent the open file limit; the `nproc` values represent the number of processes limit.

- b. Edit the `/etc/sysctl.conf` file and add following line to change the number of file descriptors.

```
fs.file-max = 26815744
```

- c. Run the following command to apply the view the settings.

```
sysctl -p
```

- d. Run the following command to verify if the value of `fs.file-max` is set to 26815744.

```
cat /proc/sys/fs/file-max
```

- e. Run the following command to view the PAM limits.

```
sudo vi /etc/pam.d/common-session
```

3. Copy the Oracle Fusion Middleware Infrastructure Installer to the middleware host and unzip it.
4. Set `JAVA_HOME`, and then start the Oracle WebLogic Server installation by using this command: `$JAVA_HOME/bin/java -jar fmw_12.2.1.4.0_infrastructure.jar`
5. Select the inventory location, such as `/scratch/biapps/installation/InvWLS` and click **OK**.
6. Select the Oracle Home location, such as `/scratch/biapps/Oracle/Middleware/ORACLE_HOME` and click **Next**.
7. Click **Next** until you complete the install steps.

Install Oracle Analytics Server

Install Oracle Analytics Server.

1. Download Oracle Analytics Server from the [Download page for Oracle Analytics Server](#). Refer to the Oracle BI Applications certification matrix in the in the Oracle Fusion Middleware Supported System Configurations page for the supported version of Oracle Analytics Server.
2. Copy the Oracle Analytics Server Installer zip file to the middleware host, and unzip the file.

3. Install Oracle Analytics Server. See instructions in Installing the Oracle Analytics Server Software.
4. Make sure you've set `JAVA_HOME`, and then run the following command to start Oracle WebLogic Server:

For Linux: `$JAVA_HOME/bin/java -jar Oracle_Analytics_Server_Linux_version-no.jar`

For Windows: `$JAVA_HOME/bin/java -jar Oracle_Analytics_Server_Windows_version-no.jar`

Where `Oracle_Analytics_Server_Linux_version-no` is the name of the Oracle Analytics Server jar file downloaded from edelivery.
5. Select the same `ORACLE_HOME` location as provided for Oracle Fusion Middleware Infrastructure, for example, `/scratch/biapps/Oracle/Middleware/ORACLE_HOME`.
6. Click **Next** until you complete the install steps.

Create Schemas Using RCU

You must run the repository creation utility (RCU) to create schemas for Oracle Analytics Server.

The RCU is available with the Oracle Fusion Middleware distribution files.

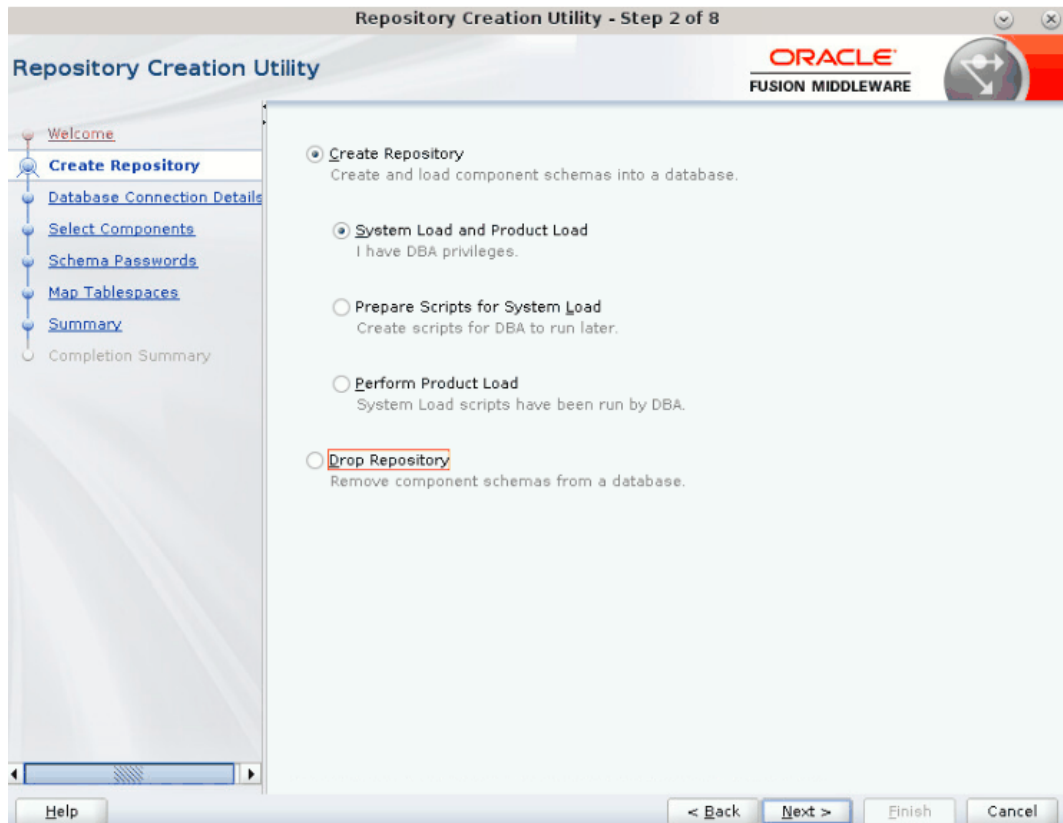
Note:

Ensure that the password is the same for all Oracle Fusion Middleware schemas such as the MDS, BIPLATFORM, STB, OPSS, IAU, and WLS schemas. Not all special characters are allowed in the schema password. It can contain #, underscore but shouldn't contain \$, parenthesis (), or @. A database service might enforce special characters, so use only underscore. The password you provide in RCU can't be changed by SQL commands.

To create the BI schema on Oracle ATP, see [Create BI Schema on Oracle ATP Database](#).

To create the BI schema, perform the following steps.

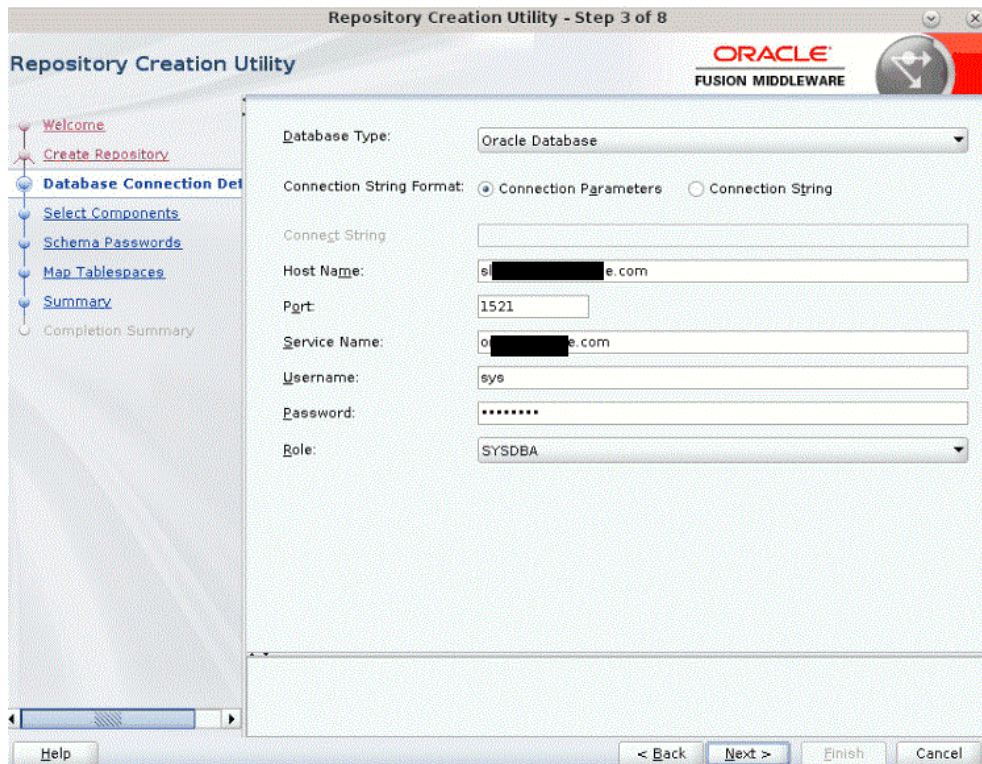
1. Start RCU using the command: `$./ORACLE_HOME/oracle_common/bin/rcu`.
You see a sequence of dialogs. Follow the instructions provided on each of the dialogs to create the required schema.
2. On the Welcome page, click **Next**.
3. Under **Create Repository**, select **System Load and Product Load**, and then click **Next**.



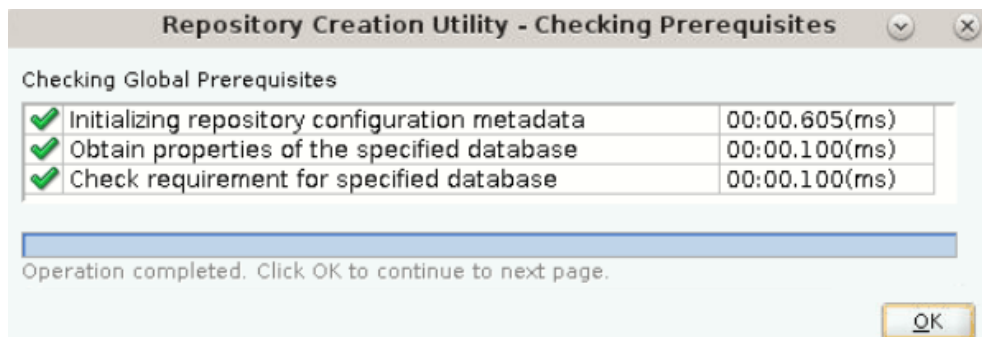
4. In the Database Connection Details dialog, enter your environment and system administrator details, and then click **Next**.

▲ Caution:

Don't use database SID while creating schemas; you must use only service name. Also, don't provide SID even in the connection string.



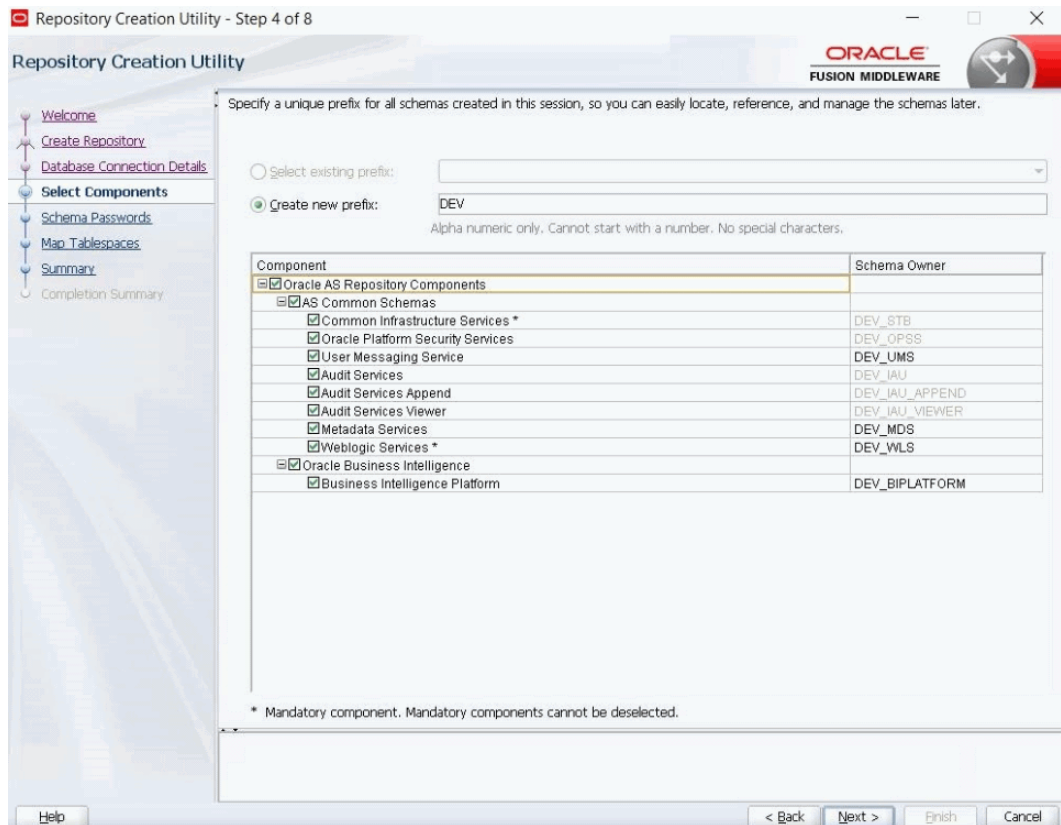
5. Wait for the RCU to complete checking the prerequisites and then click **OK**.



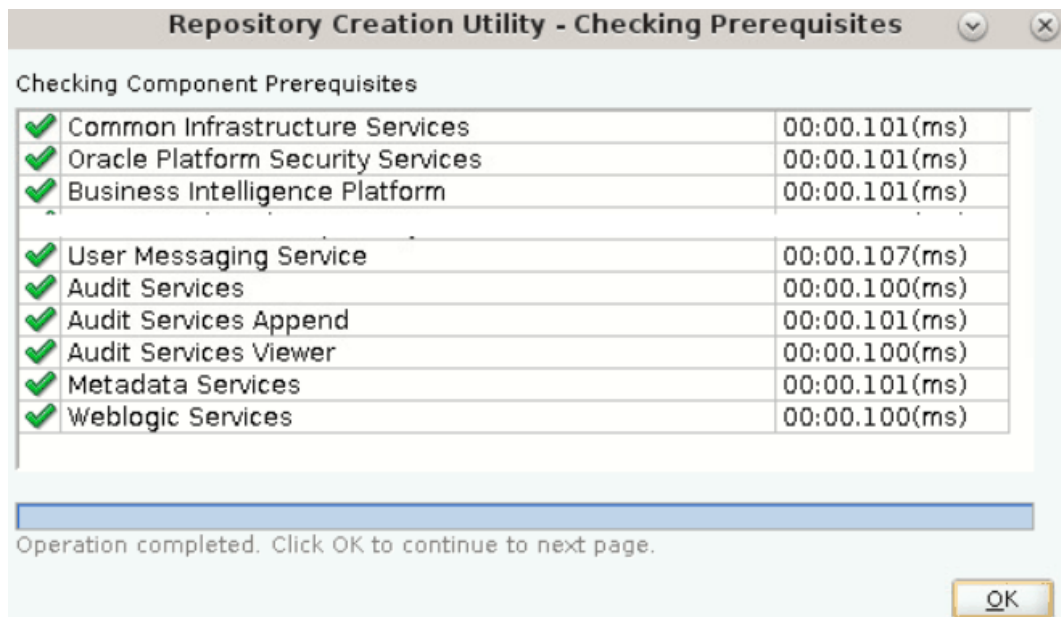
6. In the Select Components dialog, provide a schema prefix (for example, DEV or PROD or any string without special characters), select the components that you want to install, and then click **Next**.

 **Note:**

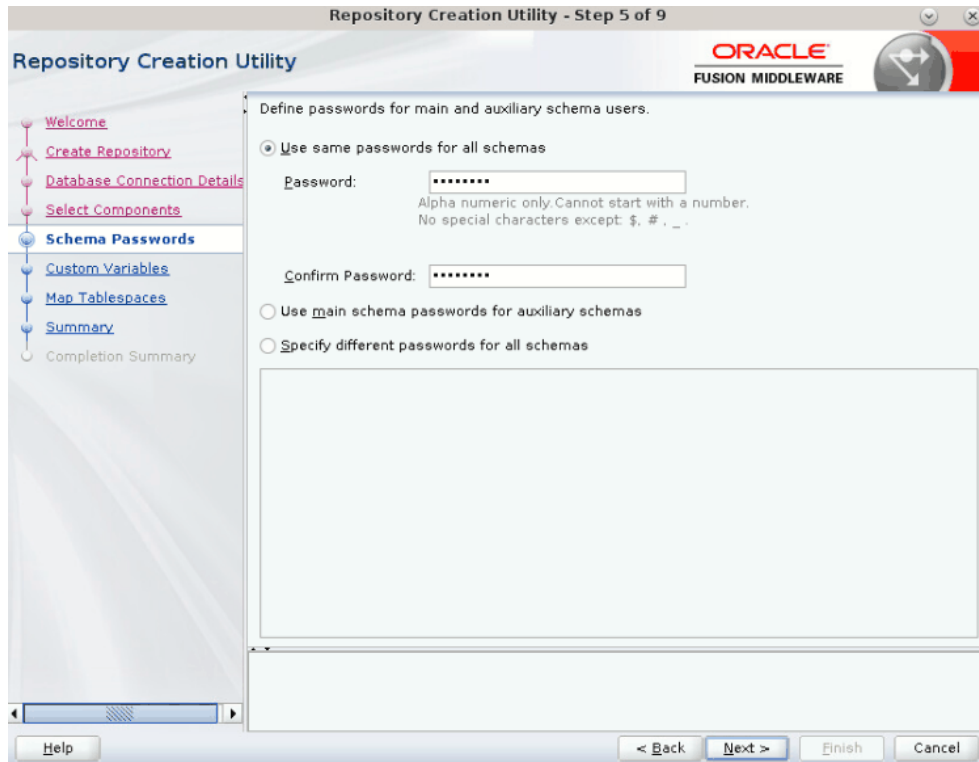
When you select Oracle Business Intelligence Platform, the RCU auto-selects other components like Common Infrastructure Services, Oracle Platform Security Services, Audit services, and Metadata Services. Don't select the ODI_REPO schema. This is created by the Oracle BI Applications RCU and is suffixed as BIA_ODI_REPO.



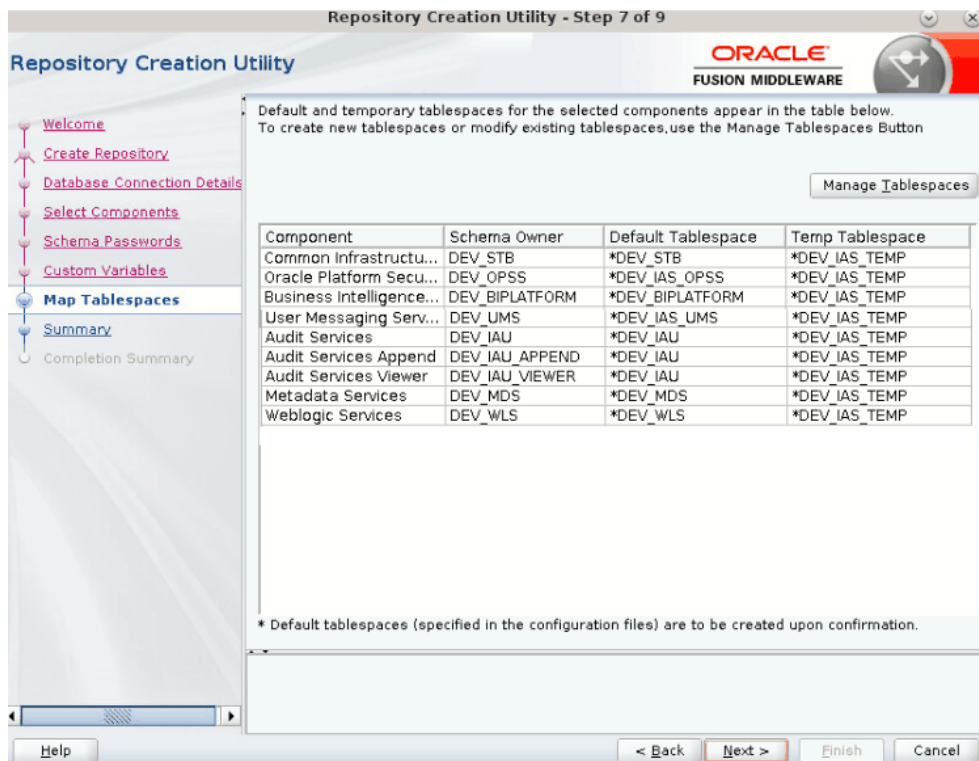
- Wait for the RCU to complete checking the prerequisites and then click **OK**.



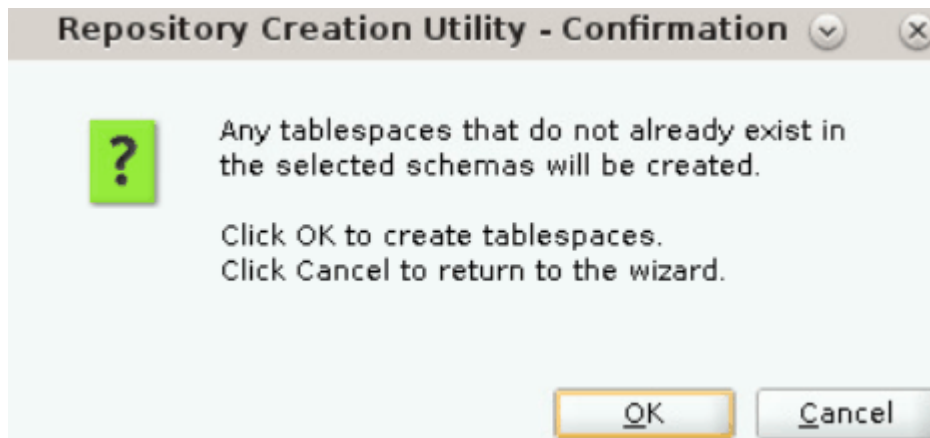
- In the Schema Passwords dialog, specify the passwords for the schema users, and then click **Next**.



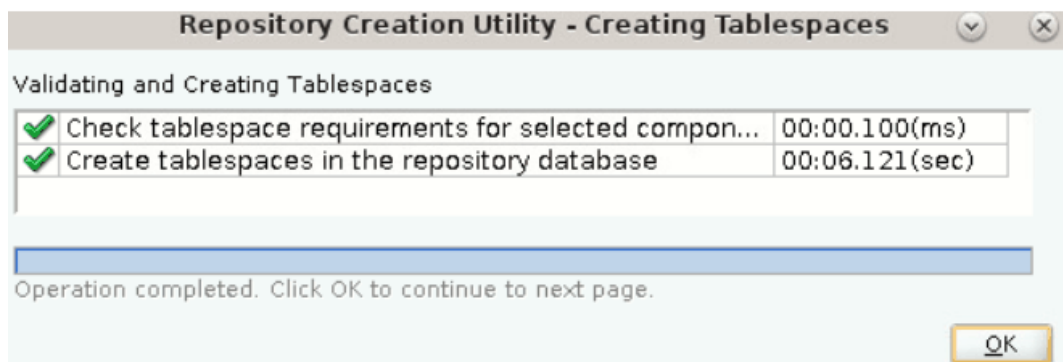
9. In the Map Tablespaces dialog, view the default and temporary tablespaces for the components that you selected. If required, create or modify tablespaces using the **Manage Tablespaces** button. Click **Next**.



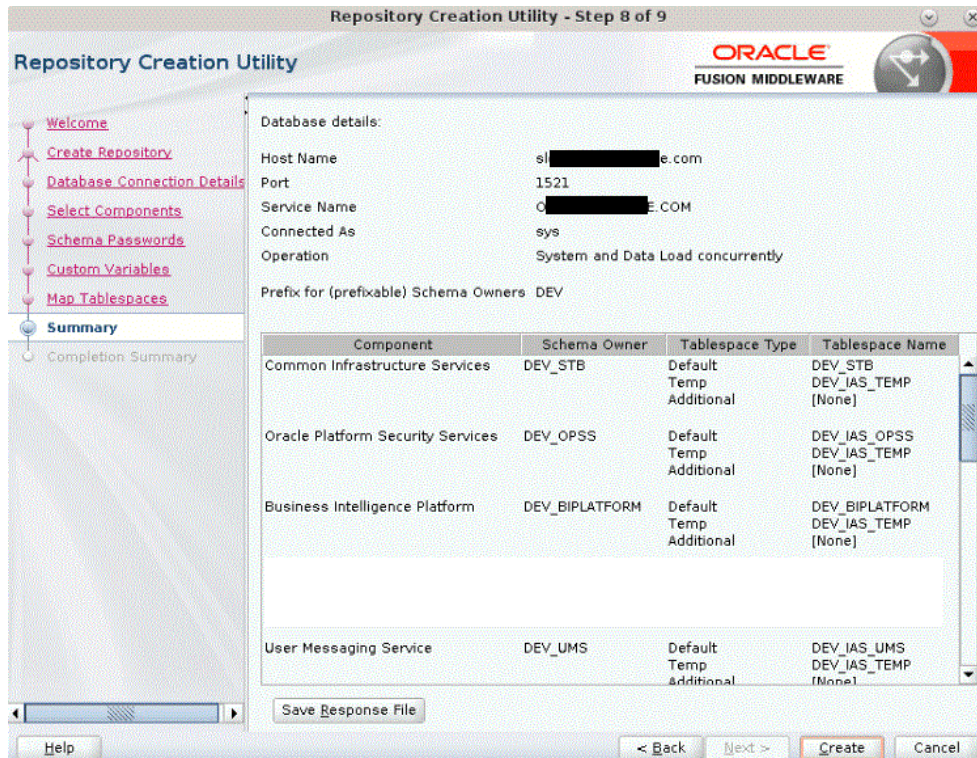
10. In the Confirmation dialog, click **OK** to create any tablespaces that don't exist in the selected schemas.



11. In the Creating Tablespaces dialog, wait for the RCU to complete creating the tablespaces, and then click **OK**.



12. In the Summary dialog, click **Create** to generate BITECH schema components.



When the schemas are created with no errors, you see the Completion Summary dialog.

13. In the Completion Summary dialog, click **Close**.

To create Oracle Business Intelligence schemas using the Oracle Business Intelligence RCU with a RAC database, see [Creating BI Schemas Using RCU with a RAC Database](#).

Install Oracle Data Integrator

Install Oracle Data Integrator using the Oracle Data Integrator installer.

1. Download Oracle Data Integrator 12.2.1.4 from the [Oracle Data Integrator Downloads](#) page.
2. Copy the Oracle Data Integrator 12.2.1.4 zip file to the middleware host, and unzip the file.
3. Make sure you have set `JAVA_HOME`, and then run the following command to start Oracle WebLogic Server:


```
$JAVA_HOME/bin/java -jar fmw_12.2.1.4.0_odi.jar
```
4. Select the same `ORACLE_HOME` location as provided for Oracle Fusion Middleware Infrastructure, for example, `/scratch/biapps/Oracle/Middleware/ORACLE_HOME`.
5. Make sure you select **Enterprise Install**.
6. Click **Next** until you complete the install steps.

Install Oracle BI Applications

Install Oracle BI Applications 11.1.1.10.3 PS4 using the installer.

1. Download the installer (jar file) from the [Oracle Business Intelligence Applications Downloads](#) page.

You see two jar files. Ensure you keep both the jar files in the same directory.

2. Prepare a text file (known as response file) using the name `biapps.resp` with the contents seen in the sample response file:

```
[ENGINE]
#DO NOT CHANGE THIS.
Response File Version=1.0.0.0.0
[GENERIC]
#provide oracle home location, where the product would be installed.
ORACLE_HOME=/scratch/biapps/Oracle/Middleware/ORACLE_HOME
```

3. Run the installer:

```
$ java -jar obia.12c.distribution-11.1.1.10.PS4-generic.jar -silent -
responseFile /tmp/biapps.resp
```

Note:

Ensure that the **responseFile** parameter value is an absolute file path and not a relative path. Use a command for the jar file which doesn't have 2 in the filename of the `responseFile`.

Create Schemas for Oracle BI Applications

You must create and load the Oracle BI Applications schemas in the database using the Repository Creation Utility (RCU).

- Make a directory in the database host and provide read/write permission to all files in that directory. For example:
 - `mkdir /dbhost/dumps`
 - `chmod 777 /dbhost/dumps`
- Copy three dump files from the middleware host to the database host folder `/dbhost/dumps`. These dumps are available under `ORACLE_HOME/oracle_common/rcu/integration/biapps/schema`.

To create the BIApps schema on Oracle Autonomous Transaction Processing (ATP), see [Create BIApps Schema on Oracle Autonomous Transaction Processing \(ATP\)](#).

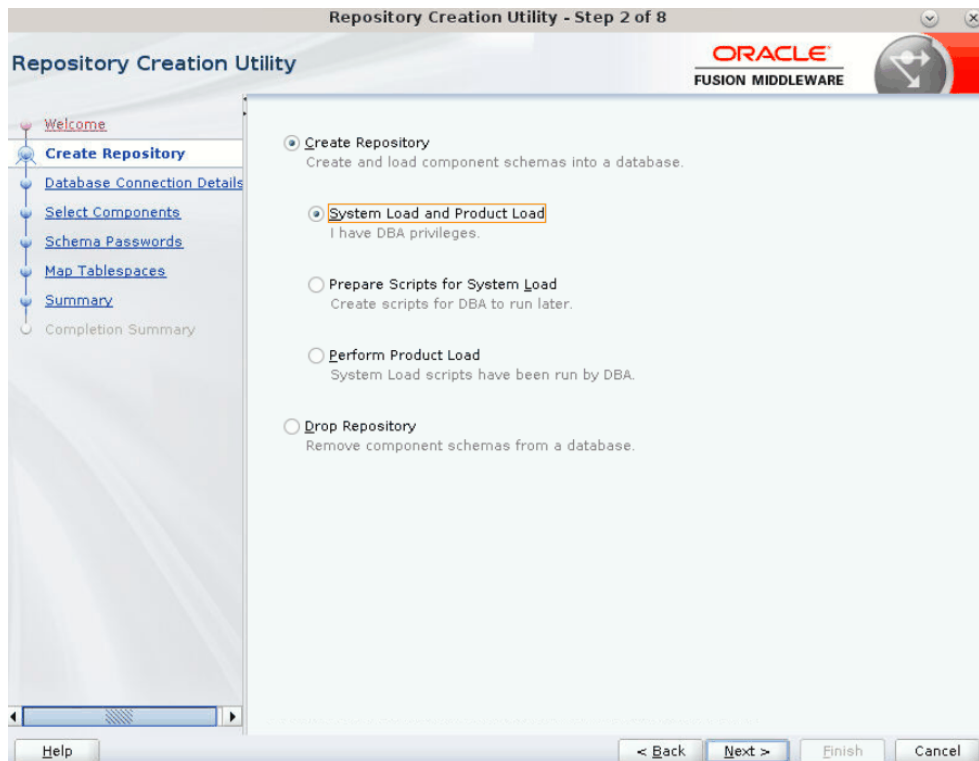
1. After installing Oracle Fusion Middleware Infrastructure, start the RCU from the `ORACLE_HOME/oracle_common/bin` directory using these commands:

```
ORACLE_HOME/oracle_common/bin/rcu -compInfoXMLLocation ORACLE_HOME/oracle_common/rcu/
config/ComponentInfo_biapps.xml -storageXMLLocation ORACLE_HOME/oracle_common/rcu/config/
Storage_biapps.xml
```

You see a sequence of dialogs. Follow the instructions provided on each dialog to create the required schema.

2. On the Welcome page, click **Next**.

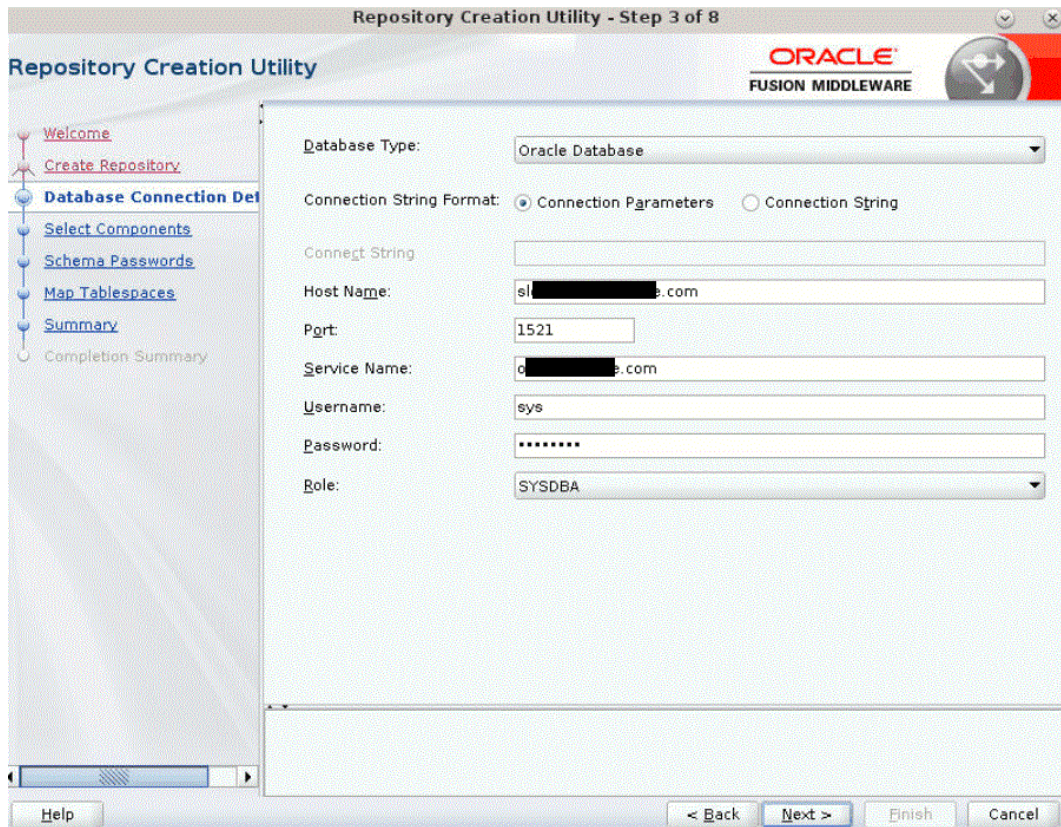
3. Under **Create Repository**, select **System Load and Product Load**, and then click **Next**.



4. In the Database Connection Details dialog, enter your database details and click **Next**. If your database is RAC, then enter the SCAN url.

Caution:

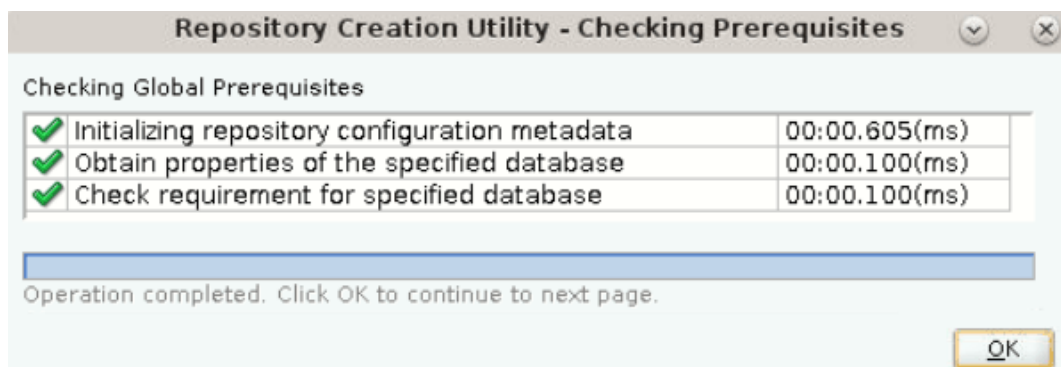
Don't use database SID while creating schemas; you must use only service name. Also, don't provide SID even in the connection string.



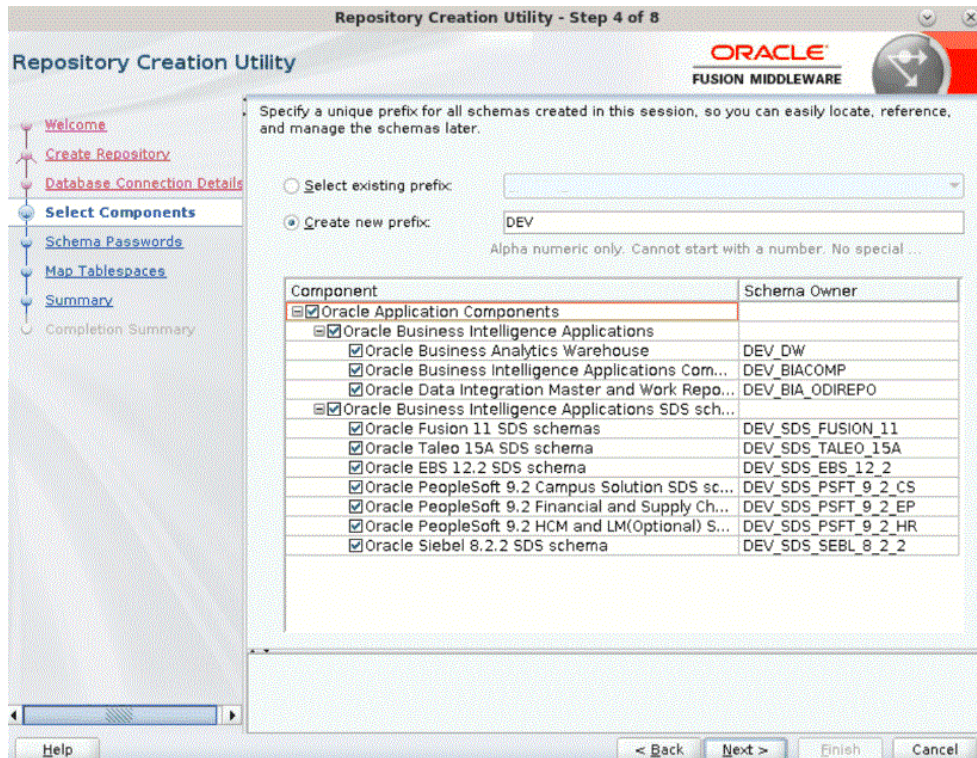
 **Note:**

If you see a warning message that the database version is more recent, then click **Ignore**.

5. Wait for the RCU to complete checking the prerequisites and then click **OK**.

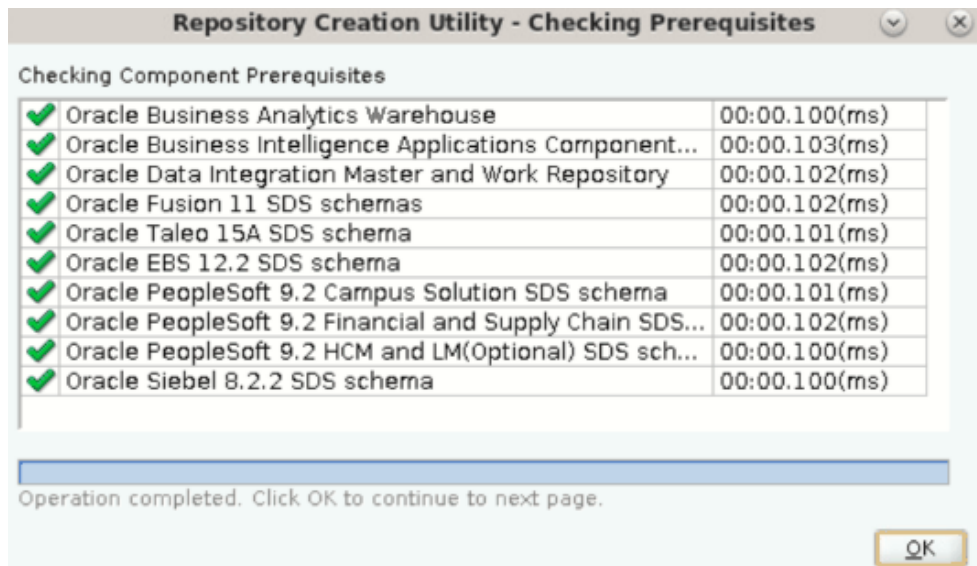


6. In the Select Components dialog, provide a schema prefix (for example, DEV or any string without special characters), select the components that you want to install, and then click **Next**.



7. Wait for the RCU to complete checking the prerequisites and then click **OK**.

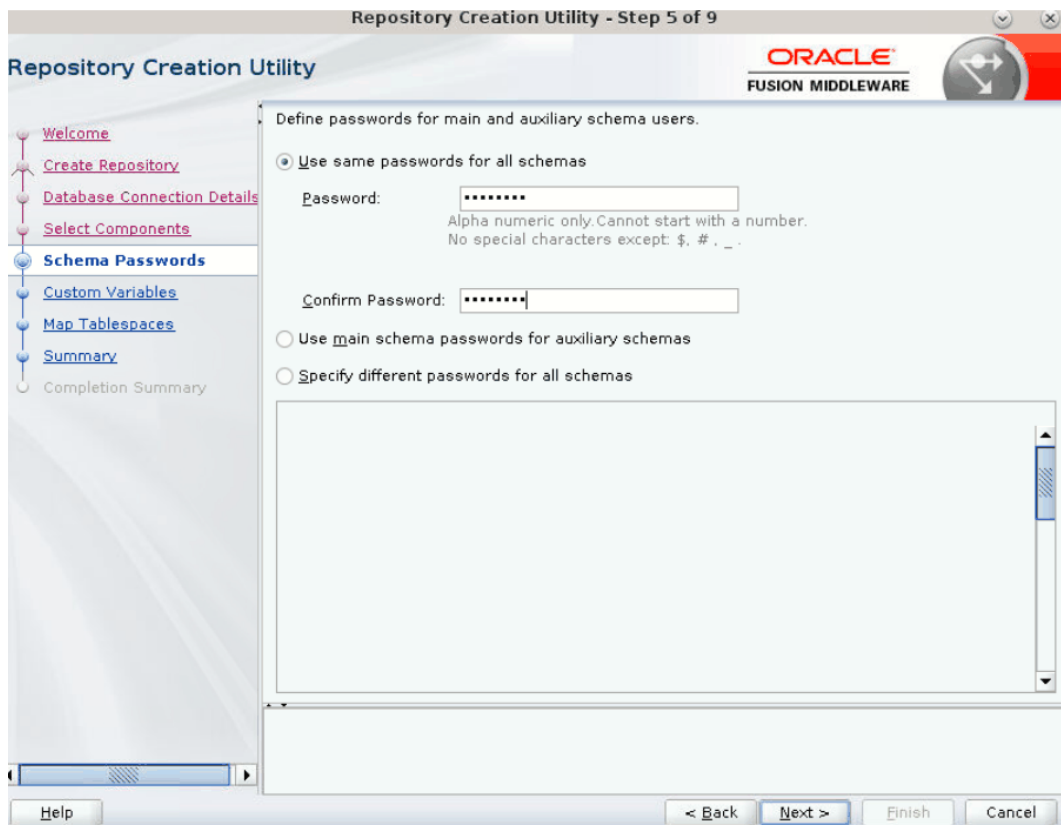
If you see a warning message stating that the version of the database is more recent than the clock, then ignore the message.



8. In the Schema Passwords dialog, specify the passwords for the schema users, and then click **Next**.

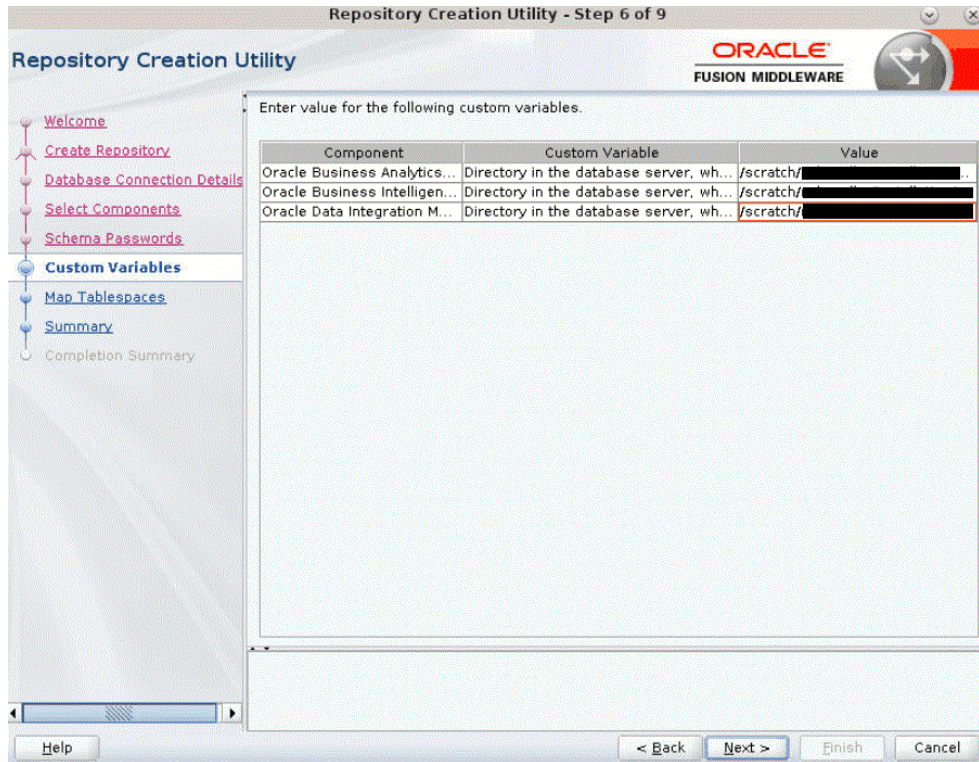
 **Note:**

Not all special characters are allowed in the schema password. It can contain number sign # or underscore _, but shouldn't contain dollar sign \$, parenthesis (), or at sign @.



9. In the Custom Variables dialog, enter the database dump location (/dbhost/dumps) for Oracle Data Integrator Master and Work repository and Oracle BI Applications components, and then click **Next**.

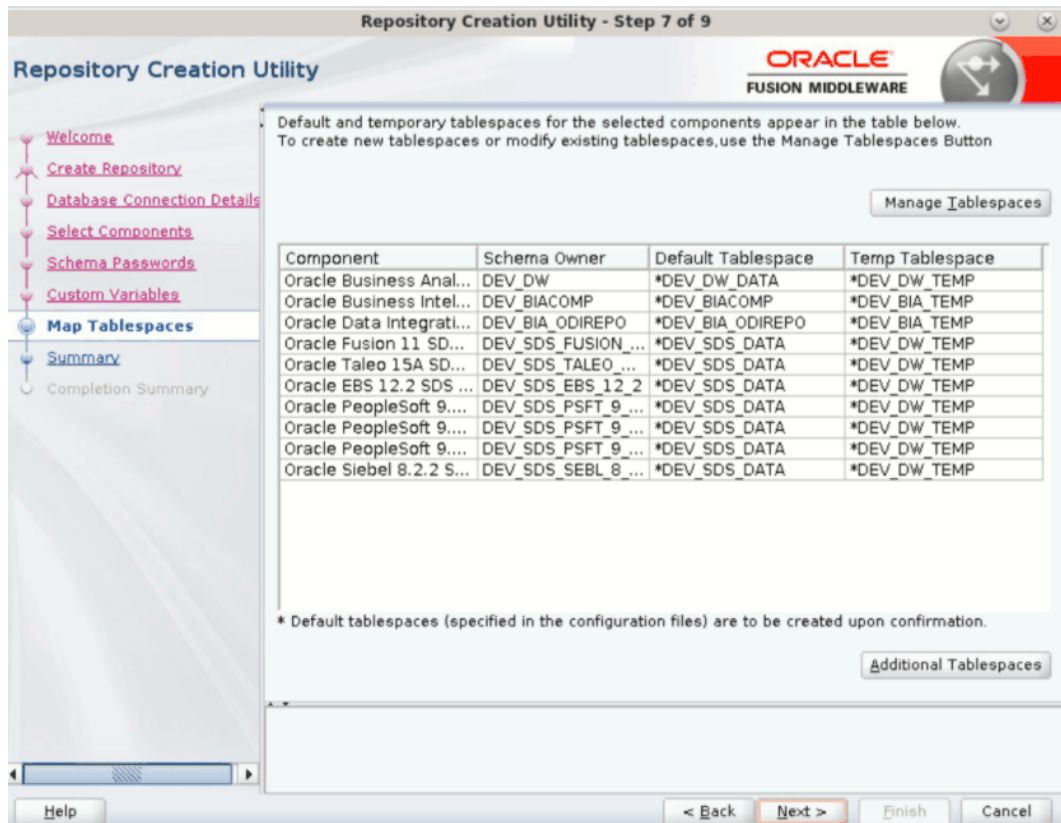
For example, copy the dump location from the machine where you installed Oracle Fusion Middleware to the database host, if the dump location is different. If all the dump files are in the same location, then specify the same dump location three times. You need to provide only the /dump location; the file name isn't required.



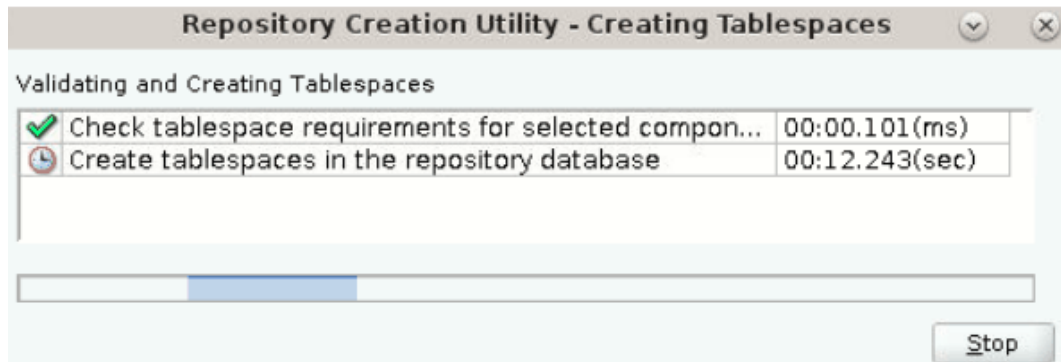
10. In the Map Tablespaces dialog, view the default and temporary tablespaces for the components that you had selected. If required, create or modify tablespaces using the **Manage Tablespaces** button after referring to the schema guidelines. Click **Next**.

 **Note:**

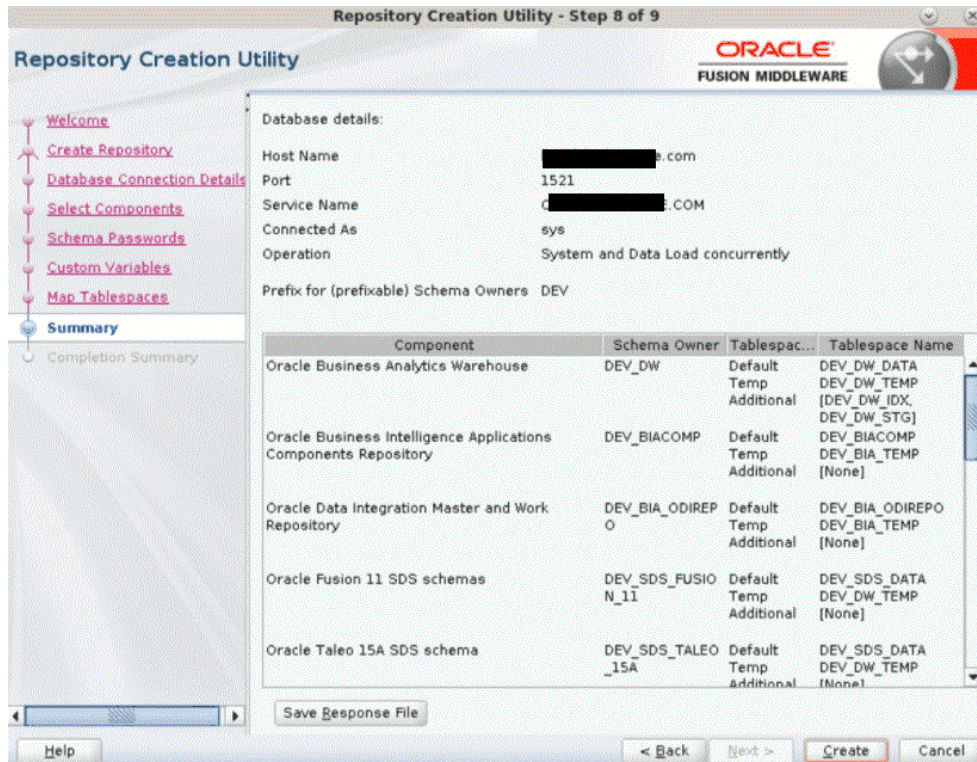
Oracle BI Applications has one schema to support all higher releases of Oracle Fusion Applications. For example, the FA_11 schema supports Oracle Fusion Applications releases 11 and higher.



- In the Creating Tablespaces dialog that is automatically launched to create any tablespaces that don't exist in the selected schemas, click **OK**.



- In the Summary dialog, click **Create** to complete the operation.



When the schemas are created with no errors, you see the Completion Summary dialog.

13. In the Completion Summary dialog, click **Close**.

Create BIApps Schema on Oracle Autonomous Transaction Processing (ATP)

Run the repository creation utility (RCU) to create the BIApps schema on Oracle Autonomous Transaction Processing (ATP).

Note:

Oracle Autonomous Transaction Processing (ATP) database provides various services consumer groups. Oracle recommends to always use “TP” ATP service name while running RCU and configuring Oracle BI Applications.

Before you run the RCU, perform these tasks:

1. Upload the `obia.dmp`, `obia_comp.dmp`, `obia_odi.dmp`, and `obia_partitions.dmp` files to Object Storage.
2. Create the ATP credentials.
3. Copy the dump files from Object Storage to the data pump directory.

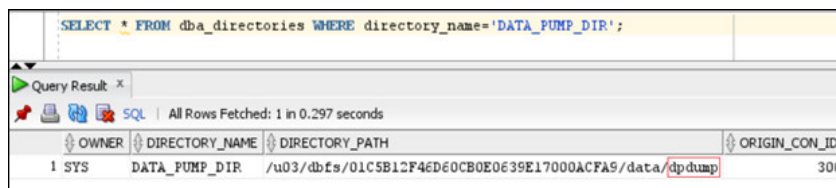
For example,

```
BEGIN
DBMS_CLOUD.GET_OBJECT(
credential_name => '<ATP CREDENTIAL NAME>',
object_uri => '<bucket URL>/Dump_file_Name',
directory_name => 'data_pump_dir');
END;
```

- Run the following SQL query on Oracle Autonomous Transaction Processing (ATP) or Oracle Autonomous Database (ADB), and note down the location name of the dump file in ATP.

```
SELECT * FROM dba_directories WHERE directory_name='DATA_PUMP_DIR';
```

Note the value of dpdump in your environment.



The screenshot shows a SQL query result in a web interface. The query is: `SELECT * FROM dba_directories WHERE directory_name='DATA_PUMP_DIR';`. The result table has four columns: OWNER, DIRECTORY_NAME, DIRECTORY_PATH, and ORIGIN_CON_ID. The data row shows: 1 SYS DATA_PUMP_DIR /u03/dbfs/01C5B12F46D60CB0E0639E17000ACFA9/data/dpdump 300. The value 'dpdump' in the DIRECTORY_PATH is highlighted with a red box.

OWNER	DIRECTORY_NAME	DIRECTORY_PATH	ORIGIN_CON_ID
1 SYS	DATA_PUMP_DIR	/u03/dbfs/01C5B12F46D60CB0E0639E17000ACFA9/data/dpdump	300

- When you run RCU in silent mode, note that when you enter the connection string in the response file, there's a space after `description=` in the wallet's `tnsnames.ora` file:

```
biaoas2023_tp = (description= (retry_count=20)
```

Remove the space after `description=` so that connection string is:

```
biaoas2023_tp = (description=(retry_count=20)
```

Perform these steps to create the BIApps schema on Oracle Autonomous Transaction Processing (ATP).

- Set the `TNS_ADMIN` environment variable to map it to the directory where the `tnsnames.ora` file is available.

For example, in Linux,

```
export TNS_ADMIN=/scratch/wallet
```

Where `/scratch/wallet` is the ATP wallet extract folder.

- Start RCU using the command:

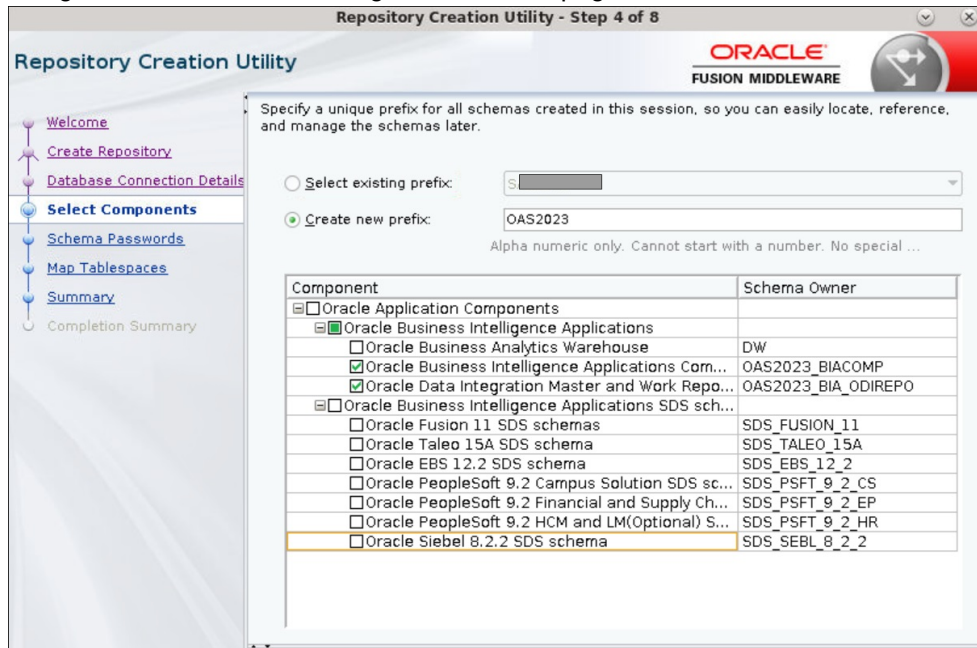
```
ORACLE_HOME/oracle_common/bin/rcu -compInfoXMLLocation /scratch/user/
installation/Middleware/Oracle_Home/oracle_common/rcu/config/
ComponentInfo_biapps_adw.xml -storageXMLLocation /scratch/user/
installation/Middleware/Oracle_Home/oracle_common/rcu/config/
Storage_biapps.xml
```

 **Note:**

The name of the component info XML file must be `ComponentInfo_biapps_adw.xml`.

3. On the Welcome page, click **Next**.
4. Under **Create Repository**, select **System Load and Product Load**, and then click **Next**.
5. Open the /scratch/wallet/tnsnames.ora file, get the connection string description for your desired service name, and enter the value for Connect String.
6. Enter Username and Password, and then click **Next**.
7. Ignore the Connection String Format alert, and then click **OK**.
8. Enter the schema prefix name, select only ODI and BIACOMP as components, and then click **Next**.

It might take some time to navigate to the next page.



9. Enter the schema passwords, and then click **Next**.
10. In the Custom Variables page, enter value as "dpdump" or the value of dpdump in your environment.
11. In the Map Tablespaces page, click **Next**.
It might take some time to navigate to the next page.
12. In the Summary page, click **Create**.

Set the BIAPPS Schema User Password as Unlimited

Oracle recommends that you ensure that the database password for all BIAPPS schema users doesn't expire. Resetting the database password might lead to failure in the BIAPPS components operations.

1. Log in to the database as `sys` user.

2. Create a profile with password-related attributes set to unlimited using the following SQL script:

```
CREATE PROFILE BIAPPS
LIMIT PASSWORD_LIFE_TIME UNLIMITED
PASSWORD_GRACE_TIME UNLIMITED
PASSWORD_REUSE_TIME UNLIMITED
PASSWORD_REUSE_MAX UNLIMITED
FAILED_LOGIN_ATTEMPTS UNLIMITED
PASSWORD_VERIFY_FUNCTION NULL
PASSWORD_LOCK_TIME UNLIMITED;
```

3. Associate all the BIAPPS users with this profile by executing the following SQL script for all the BIAPPS users:

```
ALTER USER <BIAPPS_SCHEMA_USER> PROFILE BIAPPS;
For example, ALTER USER BIAPPS_DW PROFILE BIAPPS;
```

Upgrade OPatch

Upgrade OPatch to the latest release on the `ORACLE_HOME` location where you installed Oracle Fusion Middleware Infrastructure.

1. Download the latest patches from the same location where you downloaded the Oracle BI Applications software, save the latest patches to a location, and unzip them.
2. Set `ORACLE_HOME` as the parent of product home.

For example, `setenv ORACLE_HOME /scratch/biapps/Middleware/ORACLE_HOME PATH`

3. Locate the `opatch_generic.jar` file in the folder where you have unzipped the latest patch, and apply it.

For example, `java -jar path_to_opatch_generic.jar/opatch_generic.jar -silent oracle_home=ORACLE_HOME PATH`

Apply Mandatory Patches

After the software-only install, you must apply patches on the `ORACLE_HOME` location where you installed Oracle Fusion Middleware Infrastructure 12.2.1.4.

All patches are available in the same place from where you downloaded the Oracle BI Applications software.

1. Copy the patches to a location, unzip them, and apply them using the instructions in the patch README file.
2. Set `ORACLE_HOME` as the parent of product home.

For example, `setenv ORACLE_HOME /scratch/biapps/Middleware/ORACLE_HOME`

Don't set it as `/scratch/biapps/Middleware/Oracle_home/odi`.

3. Run the `$ORACLE_HOME/OPatch/opatch apply <patch_dir>` command for all individual mandatory patches available in all the sub directories.
4. Verify each patch log and ensure that you see the "OPatch Succeeded" message.
5. Run the `$ORACLE_HOME/OPatch/opatch lsinventory` command, take output in a file, and verify that each patch number appears in that file as applied.

Configure Oracle Analytics Server

Use Configuration Assistant to configure Oracle Analytics Server.

Oracle recommends to follow About Oracle Analytics Server Installation.

In Windows, make sure that the machine name doesn't include an underscore character (`_`). If you edit the machine name to remove the underscore character, reboot the machine before you run Configuration Assistant.

1. Read and follow the step-by-step instructions in [Configure Oracle Analytics Server Domain Using Configuration Assistant](#).

 **Note:**

In Windows, when you configure the Oracle Analytics Server domain, provide the ATP connect string in the format below. Use "`\`" as the path separator.

```
CONNECT_STRING=midtier_biplatform?  
TNS_ADMIN=<ATP_WALLET_HOME>
```

For example, `CONNECT_STRING=biaoas2023_tp?TNS_ADMIN=C:\\biapps\\instantclient_19_14\\network\\admin\\atp_wallet`

2. Verify the WebLogic servers.

Run these commands, and make sure that all the components are in RUNNING state.

a. `cd DOMAIN_HOME/bitools/bin`

b. `./status.sh`

3. Run the `datamodel` command and verify that Oracle Analytics Server works.

After you configure Oracle Analytics Server, run the `datamodel.sh` (linux) or `datamodel.cmd` (Windows) command.

a. Change the directory to `DOMAN_HOME/bitools/bin`.

b. Run the following command:

```
./datamodel.sh listconnectionpool -SI <Service Instance> -U  
<WebLogic User> -S <Host Name> -N <BI Server Port> -V true
```

For example: `./datamodel.sh listconnectionpool -SI <biainstance> -U
<weblogic> -S <localhost> -N <8003> -V true`

Output example:

```
Post listconnectionpool REST with url: http://localhost:8003/bi-  
lcm/v1/si/biainstance/rpd/listconnectionpool  
{  
  "Title": "List Connection Pools",
```

```

"Conn-Pool-Info": [
  {
    "uid": "c0000000-c929-1508-ac27-0ae4a1690000",
    "connPool": "DUMMY_CP",
    "parentName": "\"DUMMY\"",
    "user": "admin1",
    "password": "706F87F7847C7A8EFED6E3F0AB261ADE
7C2E35213306F12832914CBE7A9DD95561D771DED06484112B1FC6F27B6D0D58",
    "dataSource": "testDb",
    "appServerName": "AppServer1",
    "externalizedConnection": ""
  }
],
"Variables-In-Conn-Pool": [
]
}

```

Edit Your Input File

The configuration script expects details about your environment. You must copy the `ORACLE_HOME/bi/biapps/scripts/input.txt` file to a location and edit the `input.txt` file to update the properties for your environment.

Prior to editing your `input.txt` file, ensure that you've decided your configuration plan, such as:

- If you're going to use RAC database, then see [Use a RAC Database](#).
- If you're using embedded LDAP or planning to use Oracle Internet Directory LDAP. See [Use an External LDAP](#).

Based on your choices, you need to modify the properties in your `input.txt` file. Ensure that you don't delete or comment out any of the properties.

Note:

Ensure that you edit the `input.txt` file only in Linux using the Visual Editor (Vi Editor). Don't edit the file in the Windows machine and copy to Linux. Editing in Windows causes the configuration to fail. For Windows, use the file path separator as `\\` or `.`. For example, `biprov.oracle_home.dir=c:\\Oracle\\Middleware\\Oracle_Home` or `biprov.oracle_home.dir=c:/Oracle/Middleware/Oracle_Home`. All the paths in `input.txt` should use the same format.

Property	Property Value
<code>biprov.jdk_home.dir</code>	Provide the JDK directory, under which the bin folder must exist.

Property	Property Value
<code>biprov.oracle_home.dir</code>	Provide the absolute path where you'd install Oracle BI Applications, such as <code>/u01/installation/Middleware/Oracle_home</code> . The configuration process creates a directory and you must ensure that the users have write permission on that path.
<code>biprov.db.schema.prefix</code>	Provide the schema prefix used during the Oracle Analytics Server RCU run.
<code>biprov.odi.db.schema.prefix</code>	Provide the schema prefix for the ODI schema created using the RCU.
<code>biprov.fa.admin.user</code>	Provide the administrator username for Oracle Weblogic Server. Choose any name without special characters.
<code>biprov.fa.admin.pwd</code>	Provide the password of the Oracle Weblogic Server administrator.
<code>biprov.odi.port</code>	Provide port for ODI managed server or retain the default value.
<code>biprov.odi.ssl.port</code>	Keep blank (SSL not supported in Oracle BI Applications).
<code>biprov.odi.is.ssl.configured</code>	Keep as false (SSL not supported in Oracle BI Applications).
<code>biprov.odi.user.name</code>	Provide username of Oracle BI Applications administrator. This user can access Oracle WebLogic Server, Oracle Analytics Server, Oracle Business Intelligence Applications Configuration Manager, and Oracle Data Integrator.
<code>biprov.odi.console.user.pwd</code>	Provide the password for the Oracle BI Applications administrator. Don't include special characters.
<code>biprov.odi.cluster.name</code>	Provide the cluster name for the ODI managed server or retain the default value.
<code>biprov.isRac</code>	Provide the value as false (even for scan URL).

Property	Property Value
[DW_DATABASE]	Provide the connection string in host:port/service format for the data warehouse schema.
DW_DATABASE_CONNECTION_STRING	
DW_DATABASE_SCHEMA_USER_NAME	Provide the data warehouse schema username.
DW_DATABASE_SCHEMA_PASSWORD	Provide the password for the data warehouse schema user.
DW_DATABASE_TYPE	Don't change the value.
biprov.ldap.external.user	Keep blank.
biprov.ldap.pwd	Provide a password for the LDAP administrator and note the value for future reference. The password of the embedded LDAP administrator is reset to this value.
biprov.ldap.external.url	Keep blank.
biprov.ldap.external.user_prefix	Keep blank.
biprov.ldap.external.group_prefix	Keep blank.
biprov.adw.env	Ensure that the value is set to False to disable the autonomous data warehouse configuration. In case you're planning to configure the autonomous data warehouse, refer to Configure Oracle BI Applications with Oracle Autonomous Data Warehouse .

There are other schemas, like BIACOMP, ODI_REPOSITORY, MDS, and STB. You need to fill similar sections for those schemas. After that, depending on which source you want to use, enable the appropriate SDS schemas. For example, if you're using Oracle Fusion source, then fill in the [FUSION11_SDS_DATABASE] section. For Oracle E Business Suite source, fill in [EBS122_SDS_DATABASE].

 **Note:**

For Oracle Autonomous Transaction Processing (ATP), add the database connection string in the following format:

```
ODI_REPOSITORY_DATABASE_CONNECTION_STRING=<ATP service name from  
tnsnames.ora>?TNS_ADMIN=<ATP wallet extract folder>
```

For example in Windows,

```
ODI_REPOSITORY_DATABASE_CONNECTION_STRING=biaatp_tp?TNS_ADMIN=C:\  
\biapps\instantclient_19_14\network\admin\atp_wallet
```

For example in Linux,

```
ODI_REPOSITORY_DATABASE_CONNECTION_STRING=biaatp_tp?TNS_ADMIN=  
scratch/instantclient_19_14/network/admin/atp_wallet
```

Use the same format for all the applicable schemas that you've created on ATP.

Configure Oracle BI Applications

Use Java archive `obia.config.jar` to configure Oracle BI Applications.

 **Note:**

If you want to configure Oracle BI Applications on Oracle Autonomous Data Warehouse, see [Configure Oracle BI Applications with Oracle Autonomous Data Warehouse](#).

 **Note:**

In Windows, to run the Java archive (`obia.config.jar`), launch the command prompt in Administrators Mode.

1. Edit the `input.txt` file.
2. After the `input.txt` file is ready, in the `$ORACLE_HOME/bi/biapps/scripts` directory, run the following command:

```
java -jar obia.config.jar <path of edited input.txt>/input.txt
```

Configure Oracle BI Applications with Oracle Autonomous Data Warehouse

Connect Oracle BI Applications to Oracle Autonomous Data Warehouse to access the data warehouse schemas.

My Oracle Support provides detailed instructions for obtaining the latest patches, applying the patches, and provisioning. See My Oracle Support note Doc ID 2808444.1 for instructions and details of the patch set.

Update \$ORACLE_HOME/bi/biapps/scripts/input.txt, along with the [GENERIC] and [DATABASE] sections, and update these parameters:

```
biprov.adw.env=<If new configuration run with ADW, then TRUE Else FALSE>
biprov.wallet.location=<Unzipped ADW wallet location>
adw.wallet.trustStorePassword=<ADW wallet password>
instant.client.location=<Unzipped Instant client
path>adw.service.name=<Service name, Refer to tnsnames.ora file in your
wallet. Ex: biapps_low>
adw.admin.user=<ADW admin user Ex: ADMIN>
adw.admin.pwd=<ADW admin password>
adw.default.cred.user=<Credential Name. Default text:
BIAPPS_ADW_AUTO_CREDENTIAL.>adw.default.user.name=<OCI User name. Open the
Profile menu () from OCI console and fetch User name.
Ex:oracle/first.lastname@oracle.com>
adw.auth.token=<Generate an Authentication Token. This token is required for
importing
dump from Object storage>
REMAP_SCHEMA=<BIAPPS_DW:ADW schema name for DW Ex: BIAPPS_DW:ADW_DW>
DUMP_FILE=<default_credential:<Pre-authenticated Request URL of the
Bucket>/obia.dmp Ex: default_credential:https://Server
Name/p/77Eo_TV2rtL/n/biappsdev/b/BIAPPSD_BUCKET/o/obia.dmp>
```

Make sure that the ADW wallet extracted folder contains all the files such as cwallet.sso, tnsnames.ora, and sqlnet.ora.

In the [DATABASE] section, update the database connection string for all schemas in the following format:

```
DW_DATABASE_CONNECTION_STRING=<ADW service name from tnsnames.ora?>
TNS_ADMIN=<ADW wallet extract folder>
```

Note:

Autonomous data warehouse provides various services consumer groups. Oracle recommends to always use “LOW” ADW service name while configuring Oracle BI Applications.

For example, DW_DATABASE_CONNECTION_STRING=biappsd3_low?TNS_ADMIN=/scratch/wallet

- biappsd3_low is the ADW service name from the tnsnames.ora file. Check your service name.
- /scratch/wallet is the folder where you have extracted the ADW wallet zip and it contains files such as cwallet.sso, tnsnames.ora, and sqlnet.ora .

Also, update the required SDS schema details.

Use the following commands to run the configuration:

 **Note:**

In Windows, to run the Java archive (obia.config.jar), launch the command prompt in Administrators Mode.

```
cd $ORACLE_HOME/bi/biapps/scripts/  
java -jar obia.config.jar input.txt
```

Load Translation Data

Load translation data into the BIACOMP schema of Oracle BI Applications.

 **Note:**

You must complete all the steps listed in [Installing and Setting Up Oracle BI Applications](#) without any errors prior to loading the translation data.

If you want to load translation data on ATP, see [Load Translation Data on Oracle Autonomous Transaction Processing \(ATP\)](#).

Ensure that the:

- Database is up and running.
 - BIACOMP schema is available.
 - The C_RPD_MSGS, C_RPD_MSGS_REL, and C_RPD_MSGS_TL tables exist in the BIACOMP schema.
1. In your middleware host, navigate to <ORACLE_HOME>/bi/nls.
 2. Copy the file rpd_translations.zip to the database host.
 3. In your database host, complete the following:
 - a. Unzip the translations file rpd_translations.zip in your database host and navigate to the **rpd_translations** directory.
 - b. Run the script:
 - For Linux/Unix:
load0tbiSeeddataSqlldr.sh thrice to load translations data into three tables using the following command:

```
sh load0tbiSeeddataSqlldr.sh <DB user name> <DB password>  
<db Connect String> <Control File> <Log  
Directory> <CSV File Location> <Oracle DB Home Path>
```

For example:

```
sh load0tbiSeeddataSqlldr.sh BIAPPS_BIACOMP welcome1  
dbhost.mycomany.com:1616/DBServiceName
```

```
importcrpdmsgs_ctl /u01/logs c_rpd_msgs_delim.csv
/scratch/db/DB_Home
```

```
sh load0tbiSeeddataSqlldr.sh BIAPPS_BIACOMP welcome1
dbhost.mycomany.com:1616/DBServiceName
  rpd_translations/importcrpdmsgsrel_ctl /u01/logs
c_rpd_msgs_rel_delim.csv
  /scratch/db/DB_Home
```

```
sh load0tbiSeeddataSqlldr.sh BIAPPS_BIACOMP welcome1
dbhost.mycomany.com:1616/DBServiceName
  importcrpdmsgstl_ctl /u01/logs c_rpd_msgs_tl_delim.csv
  /scratch/db/DB_Home
```

- **For Windows:**

```
<DB_OH>/bin/sqlldr <BIACOMP USER>/<BIACOMP password>@<dbtns>
data=<path of file>\c_rpd_msgs_delim.csv control=<path of
ctl>\importcrpdmsgs_ctl log=c:\temp\crpdmsgs.log
```

```
<DB_OH>/bin/sqlldr <BIACOMP USER>/<BIACOMP password>@<dbtns>
data=<path of file>\c_rpd_msgs_rel_delim.csv control=<path of
ctl>\importcrpdmsgsrel_ctl log=c:\temp\crpdmsgsrel.log
```

```
<DB_OH>/bin/sqlldr <BIACOMP USER>/<BIACOMP password>@<dbtns>
data=<path of file>\c_rpd_msgs_tl_delim.csv control=<path of
ctl>\importcrpdmsgstl_ctl log=c:\temp\crpdmsgstl.log
```

4. After successfully running the script `load0tbiSeeddataSqlldr.sh`, verify that the `C_RPD_MSGS`, `C_RPD_MSGS_REL`, and `C_RPD_MSGS_TL` tables have the translations data.

Load Translation Data on Oracle Autonomous Transaction Processing (ATP)

Load translation data into the BIACOMP schema of Oracle BI Applications.

1. Before you load translation data into the BIACOMP schema of Oracle BI Applications, make sure that:
 - The database is up and running.
 - The BIACOMP schema is available.
 - The `C_RPD_MSGS`, `C_RPD_MSGS_REL`, and `C_RPD_MSGS_TL` tables exist in the BIACOMP schema.
 - The steps listed in [Install and Configure Oracle BI Applications](#) are completed without any errors.
2. In your middleware host, navigate to `<ORACLE_HOME>/bi/nls`.
3. Unzip the `rpd_translations.zip` translations file in your OCI compute machine and navigate to the `rpd_translations` directory.

Run the script:

- For Linux/Unix, run the `loadOtbiSeeddataSqlldr.sh` script thrice to load translations data into three tables using the following command:

```
sh loadOtbiSeeddataSqlldr.sh <DB user name> <DB password>
<atp_service_name> <Control File> <Log Directory>
<CSV File Location> <Instant Client loc>
```

For example:

```
<rpd translation unzip path>/rpd_translations/
loadOtbiSeeddataSqlldr.sh DEV_BIACOMP <DB password> biaatp_tp
<rpd translation unzip path>/rpd_translations/
importcrpdmsgs.ctl /u01/biapps/log_msgs <rpd translation unzip
path>/rpd_translations/c_rpd_msgs_delim.csv /u01/biapps/
instantclient_21_4
```

```
<rpd translation unzip path>/rpd_translations/
loadOtbiSeeddataSqlldr.sh DEV_BIACOMP <DB password> biaatp_tp
<rpd translation unzip path>/rpd_translations/
importcrpdmsgsrel.ctl /u01/biapps/log_msgs_rel <rpd translation
unzip path>/rpd_translations/c_rpd_msgs_rel_delim.csv /u01/
biapps/instantclient_21_4
```

```
<rpd translation unzip path>/rpd_translations/
loadOtbiSeeddataSqlldr.sh DEV_BIACOMP <DB password> biaatp_tp
<rpd translation unzip path>/rpd_translations/
importcrpdmsgstl.ctl /u01/biapps/log_msgs_tl <rpd translation
unzip path>/rpd_translations/c_rpd_msgs_tl_delim.csv /u01/biapps/
instantclient_21_4
```

- For Windows:

```
<DB_OH>/bin/sqlldr <BIACOMP USER>/<BIACOMP
password>@<atp_service_name> data=<path of
file>\c_rpd_msgs_delim.csv control=<path of
ctl>\importcrpdmsgs.ctl log=c:\temp\crpdmsgs.log
```

```
<DB_OH>/bin/sqlldr <BIACOMP USER>/<BIACOMP
password>@<atp_service_name> data=<path of
file>\c_rpd_msgs_rel_delim.csv control=<path of
ctl>\importcrpdmsgsrel.ctl log=c:\temp\crpdmsgsrel.log
```

```
<DB_OH>/bin/sqlldr <BIACOMP USER>/<BIACOMP
password>@<atp_service_name> data=<path of
file>\c_rpd_msgs_rel_delim.csv control=<path of
ctl>\importcrpdmsgstl.ctl log=c:\temp\crpdmsgstl.log
```

4. After you successfully run the `loadOtbSeeddataSqlldr.sh` script, verify that the `C_RPD_MSGS`, `C_RPD_MSGS_REL`, and `C_RPD_MSGS_TL` tables have the translations data.

Configure Oracle Data Integrator Studio

Use Oracle Data Integrator Studio (ODI Studio) to manage and configure Oracle Data Integrator (ODI).

ODI Studio is installed as part of the standalone or enterprise option under the specified `ORACLE_HOME` directory. ODI Studio provides four graphical navigators for managing ODI artifacts: Designer, Operator, Topology, and Security. ODI Studio can also directly execute jobs on demand, and you can use it for development and testing.

Configuring Authentication

Use these instructions to configure authentication.

1. If you're using embedded LDAP, then go to `DOMAIN_HOME/odi-client-config/embedded` directory. In case you're using Oracle Internet Directory as LDAP, then browse the `DOMAIN_HOME/odi-client-config/external` directory.
2. Copy `cwallet.sso` and `jps-config.jse.xml` to `ORACLE_HOME/odi/studio/bin`.
3. Copy `DOMAIN_HOME/config/fmwconfig/System-jazn-data.xml` to the `ORACLE_HOME/odi/studio/bin` directory.

Setting the JPS Config File Location

Use these instructions to set the location of the JPS Config file.

1. Navigate to the `ORACLE_HOME/odi/studio/bin` directory.
2. Open the `odi.conf` file in an editor.
3. At the bottom of the file, add the following:

```
AddVMOption -Doracle.security.jps.config=./jps-config-jse.xml
AddVMOption -Dexclude.technology.refresh.seeded.objects=true
```

4. Save the file.

Starting Oracle Data Integrator Studio

Use these instructions to start ODI Studio.

1. Navigate to the `ORACLE_HOME/odi/studio` directory.
2. Enter the command for Linux `./odi.sh -clean -initialize`.

Connecting to the Master Repository

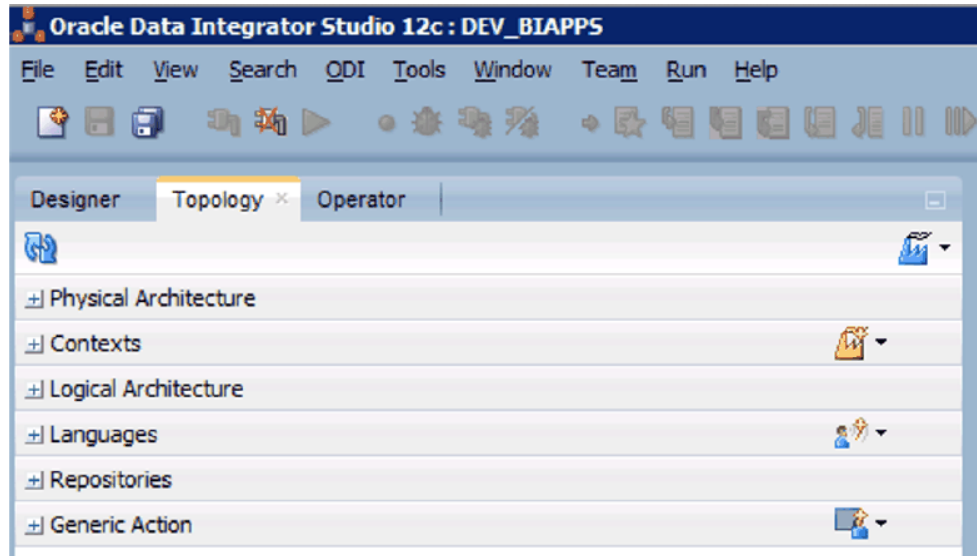
Use these instructions to create an ODI Repository Login and connect to the master repository using the newly created login in Oracle Data Integrator.

1. Sign in to ODI Studio, from the menu, select **File**, then select **New**.

2. In the New Gallery dialog, select **ODI** under **Categories** in the left pane, select **Create a New ODI Repository Login** under **Items** in the right pane, and then click **OK**.
3. In the Repository Connection Information dialog, provide the following details:
 - a. In the Oracle Data Integrator Connection section, enter:
 - **Login Name:** Specify a custom login name.
 - **User:** Specify a user who has supervisor privileges. For example, user specified as `biprova.odi.user.name` in the `input.txt` file.
 - **Password:** Specify the password for the user you entered in the **User** field.
 - b. In the Database Connection (Master Repository) section, enter:
 - **User:** Specify the schema user name for the Master repository. This must be `prefix_BIA_ODIREPO` as specified in the Select Components dialog in RCU.
 - **Password:** Specify the schema password for the Master repository. This must be the password that you specified in the Schema Passwords dialog in RCU.
 - **Driver List:** Select the appropriate driver for your database from the list of values.
 - **URL:** Specify the connection URL. Click on the magnifying glass icon for more information about the connection details for your driver.
 - c. In the Work Repository section, select **Master Repository Only**.

4. Click **Test** to test the connection, and fix any errors. After the test is successful, click **OK** to create the connection.
5. In the New Wallet Password dialog, specify and confirm a new wallet password.
After you have successfully created a new login, you are returned to ODI Studio.
6. In ODI Studio, under the Designer tab, select **Connect to Repository** and, when prompted, provide your new wallet password.
After you provide your wallet password, you see the Oracle Data Integrator Login dialog.
7. Provide the following information to log in:
 - a. Select the name of the new login that you created from the list of values.
 - b. Specify a user who has supervisor privileges.
 - c. Provide the password for the user who has supervisor privileges.

After successfully signing into ODI, you see the ready-to-use ODI repository artifacts.



Regenerating the Security Files

Administrator must regenerate the security files and redistribute the files to all instances of ODI Studio if there's a change in the password of Oracle BI Applications administrator or any Oracle Data Integrator user.

1. Execute the `wlst.sh` script.

```
ORACLE_HOME/oracle_common/common/bin/wlst.sh ORACLE_HOME/bi/biapps/
scripts/createJPSArtifactsODI.py embedded --ADMIN_USER_NAME
<WebLogic Server Admin User> --DOMAIN_HOSTNAME <Hostname> --
DOMAIN_PORT <WebLogic Port> --DOMAIN_HOME_PATH DOMAIN_HOME
```

The updated JPS configuration file and credential wallet are created in the `DOMAIN_HOME/odi-client-config/embedded` directory.

2. On all the ODI Studio instances, copy the updated JPS configuration file and the credential wallet files from the `DOMAIN_HOME/odi-client-config/embedded` directory to the `ORACLE_HOME/odi/studio/bin` directory.
3. Make sure that the `odi.conf` file is edited correctly to configure user access for ODI Studio.

The `odi.conf` file is located in the `ORACLE_HOME/odi/studio/bin` directory.

See [Setting the JPS Config File Location](#).

Use a RAC Database

If you're planning to use a real application cluster (RAC) database to configure Oracle BI Applications, then follow these additional steps.

The configuration flow is largely the same whether you're using a non-RAC database or a RAC database. The instructions to install Oracle Fusion Middleware Infrastructure, Oracle Business Intelligence, creating Oracle Business Intelligence

schemas, installing Oracle Data Integrator, and installing Oracle BI Applications are the same whether you're using a non-RAC database or a RAC database. For creating schemas using the Oracle Business Intelligence RCU when you've a RAC database, see [Creating BI Schemas Using RCU with a RAC Database](#).

 **Note:**

Oracle recommends that you use the SCAN URL for the RAC database. You can directly use the SCAN URL in the configuration.

1. While editing the input file:
 - a. Keep the value of **biprov.isRAC** parameter as **False**.
 - b. Provide the SCAN URL for all the database schemas in the input file.
 - c. Keep **DATABASE_TYPE** as **Oracle Database**.
2. Run `config` to complete the process.

Use an External LDAP

Instead of using Oracle Weblogic's embedded LDAP, you can use Oracle Internet Directory as an external LDAP for users.

 **Note:**

Oracle BI Applications only supports Oracle Internet Directory.

If you want to configure Oracle BI Applications with Oracle Internet Directory, first configure Oracle BI Applications to use embedded LDAP, and then see the instructions in My Oracle Support Doc ID 2818769.1 to switch from embedded LDAP to Oracle Internet Directory.

Configure Oracle BI Applications for High Availability

You can configure Oracle BI Applications in high availability mode or horizontal scale-out mode. Two physical hosts support in domain and both installs have Oracle WebLogic Server running. The Oracle HTTP server is the load balancer.

To configure high availability, you must review the install configuration step of the single node installation and complete a few additional steps. You may either choose to set up high availability manually or use the automated scripts. The high-level steps to set up high availability are:

1. Install and configure Oracle BI Applications on Node1. Ensure all components are up and running.
2. Install Oracle BI Applications on Node2. Perform software only installation including prerequisite patches and don't run config.
3. Configure high availability for Oracle Analytics Server.
4. Configure high availability for Oracle Data Integrator.
5. Configure DSN for high availability on Windows.

6. Perform post-configure steps for high availability.
7. Configure Oracle HTTP Server (OHS) for high availability.

 **Note:**

You can perform steps 3 and 4 either manually or by using automated scripts.

Machine or host name longer than 32 characters isn't supported. Specify a shorter name.

Ensure the following prerequisites:

1. Set up shared files and directories. For data sharing, you need a common mount point where both host can read/write (this isn't `ORACLE_HOME`). See Set Up Shared Files and Directories.
2. Node1 and Node2 hosts are reachable from each other. You must be able to ping the hosts from one another.
 - a. The `/etc/hosts` file on each host must have entry of the other host.
 - b. The IP address obtained with a ping command and the IP entry in the `/etc/hosts` file must be the same.
3. In Node1, install JDK, Oracle Fusion Middleware Infrastructure (Oracle WebLogic Server), Oracle Analytics Server, Oracle Data Integrator, and Oracle BI Applications and complete the configuration. Node1 must be working as single node installation.
4. The same path is used for the `Oracle_Home` on both nodes. For example, if `ORACLE_HOME` in node1 is `/u01/middleware` then node2 should also have `ORACLE_HOME` in `/u01/middleware`. But `ORACLE_HOME` mustn't be the common drive between these hosts. See Add New Computers.
5. In Node2, complete a software-only installation of JDK, Oracle Fusion Middleware Infrastructure, Oracle Analytics Server, Oracle Data Integrator and Oracle BI Applications. Don't run configuration step in Node2. Node2 has the same path of `ORACLE_HOME` from root as Node1 has.

Configure High Availability with Automated Scripts

Use these steps to configure high availability with automated scripts.

1. Update your `input.txt` file available at `<Oracle_Home>/bi/biapps/scripts/input.txt` with these sample parameters:
 - `biprov.ha.bi_domain.dir=<DOMAIN_HOME>`
 - `biprov.ha.shared.dir=<HA_SHARED_PATH>`
 - `biprov.ha.node2.hostname=<NODE2_HOST_NAME>`. For example, `biprov.ha.shared.dir=/shared_mount/nfs_shared` and `biprov.ha.node2.hostname=node2.mycompany.com`
2. Execute the following script on node1 to clone machine and bi server and update global cache settings:

```
<Oracle_Home>/bi/biapps/scripts/configure_HA_primary.sh  
<INPUT_TXT_FILE_PATH>.
```

3. Execute the following script on node2 to create cloned biadomain and start nodemanager:

```
<Oracle_Home>/bi/biapps/scripts/configure_HA_secondary.sh  
<INPUT_TXT_FILE_PATH>.
```

4. Execute the following script on node1, to create ODI_server2 server and start all components:

```
<Oracle_Home>/bi/biapps/scripts/configure_HA_primary_ODI.sh  
<INPUT_TXT_FILE_PATH>.
```

5. Verify that all components are up and running. For example, `$DOMAIN_HOME/bitools/bin/status.sh`.

Configuring High Availability with Manual Steps

Use these steps to configure high availability manually.

Before you proceed, ensure that you've completed these steps:

1. Install and configure Oracle BI Applications on Node1. Ensure all components are up and running.
2. Install Oracle BI Applications on Node2. Don't run config. You must perform the software-only install.

For Oracle Analytics Server high availability steps, see [About Scaling Oracle Analytics Server](#) and for Oracle Data Integrator high availability steps, see the Oracle Support document "How To Create And Deploy a High Availability ODI 12c Java EE Agent to a Cluster on WebLogic Server" (Doc ID 1676176.1).

1. On Node 1, complete the following steps to change the Singleton Data Directory (SDD). See [Change the Singleton Data Directory \(SDD\)](#).

 **Note:**

Be sure install and config is completed in node1, and node1 is up and running.

- a. Export the `$ORACLE_HOME` and `$DOMAIN_HOME` environment variables. For example, `export ORACLE_HOME=/u01/Middleware/Oracle_Home` and `export DOMAIN_HOME=/u01/user_projects/domains/biadomain`.
- b. Open a shell session in node 1 and stop all components using the `stop.sh` command.
- c. Navigate to the `$DOMAIN_HOME/config/fmwconfig/bienv/core` directory.
- d. Make a backup of the `bi-environment.xml` file and edit it to change the Singleton Data Directory (SDD). Ensure that the new SDD points to a shared directory that can be accessed by both nodes of the cluster. Your mount point or share must persist through reboots;. For example, `/shared_mount/nfs_shared/bidata`.

- e. Copy the `bidata` directory to the new SDD location. For example,

```
cd <DOMAIN_HOME>
cp -R bidata /shared_mount/nfs_shared/bidata
```

- f. Start all components on node 1 using the `start.sh` command.
- g. Set up the global cache using the sample command `Create directory /shared_mount/nfs_shared/cache`.
- i. Sign in to Oracle Enterprise Manager Fusion Middleware Control.
 - ii. Click **Target Navigation**, expand **Business Intelligence**, select **biinstance**, click the Configuration tab, and then the Performance subtab.
 - iii. Click **Lock & Edit** available under **Lock** icon and provide the global cache path and size.
 - iv. Click **Apply** button and then **Activate Changes** available under the **Lock** icon.
- h. Start all components to make sure that the changes are done correctly using the `start.sh` command.
- i. Stop all components on the node1 using the `stop.sh` command.
- j. Create BI machine and BI server for node2 using the clone command:

```
DOMAIN_HOME/bitools/bin/clone_bi_machine.sh|cmd [-m <new computer name>] <listen address> <Jar file>
```

For example:

```
ORACLE_HOME/user_projects/domains/bi/bitools/bin/
clone_bi_machine.sh example.com /oracle/nfs_shared/clone/
demobi2.jar
```

2. On Node 2, complete the following steps:

 **Note:**

Oracle BI Applications is installed on Node2 but `config` isn't yet executed.

- a. Create the domain directory, as it exists in the node. Use the same directory and naming structure such as `mkdir -p /u01/user_projects/domains/biadomain`. Ensure that the `DOMAIN_HOME` in both nodes specifies the same path name.
- b. Export the `$ORACLE_HOME` and `$DOMAIN_HOME` environment variables.
- c. Unpack the domain by using the template jar that you created using the command: `ORACLE_HOME/oracle_common/common/bin/unpack.sh -template=[location of copied jar file from primary computer] -domain=DOMAIN_HOME -nodemanager_type=PerDomainNodeManager`

For example:

```
Oracle_Home/oracle_common/common/bin/unpack.sh -template=/
refresh/home/oracle/nfs_shared/clone/demobi2.jar -domain=../../..
```

```
Oracle_Home/user_projects/domains/bi -
nodemanager_type=PerDomainNodeManager
```

- d. On the new computer, start the Node Manager and Managed Server to create the config.xml file.

For example:

```
Oracle_Home/user_projects/domains/bi/bin/startNodeManager.sh
```

3. On Node 1 complete the following steps:

- a. Start the components AdminServer, bi_server1, ODI_server1, obis1, obips1, obiccs1, obijh1, obisch1, and bi_server2 using the `$DOMAIN_HOME/bitools/bin/start.sh -i "AdminServer,bi_server1,ODI_server1,obis1,obips1,obiccs1,obijh1,obisch1,bi_server2"` command.

- b. Start the components obis2, obips2, obiccs2, obijh2, and obisch2 using the `$DOMAIN_HOME/bitools/bin/start.sh -i "obis2,obips2,obiccs2,obijh2,obisch2"` command.

- c. Re-synchronize the data source on the new computer.
On the new computer, run the following script:

```
DOMAIN_HOME/bitools/bin/sync_midtier_db.sh
```

- d. On the primary computer, start the new system components for the new computer:
Change directory to `DOMAIN_HOME/bitools/bin/`.

Enter:

```
./start.sh -i obis2,obips2,obiccs2,obijh2,obisch2
```

If you're configuring high availability in Windows, then you must create and configure DSN. Refer to the DSN Creation section to complete those steps and then continue to configuring ODI for high availability.

Configuring High Availability for ODI

Use these steps to configure high availability for ODI.

- On Node 1 complete the following steps:
 1. Create ODI_server2 and assign it to the machine and cluster using these `wlst` commands:

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

```
connect('<WLS_USER>', '<WLS_USER_PASSWORD>',
'<WLS_ADMIN_SERVER_HOST>:<WLS_ADMIN_SERVER_PORT>')
edit()
startEdit()
cd('/')
cmo.createServer("ODI_server2")
cd('/Servers/' + "ODI_server2")
cmo.setListenAddress(<node2_host_name>)
cmo.setListenPort(<node2_odi_port>)
activate()
```

Bring down all components by executing `stop.sh` and then connect to `wlst` in offline mode using:

```
$ORACLE_HOME/oracle_common/common/bin/wlst.sh
readDomain(<DOMAIN_HOME>)
assign('Server', 'ODI_server2', 'Machine', <node2_host_name>)
assign('Server', 'ODI_server2', 'Cluster', <odi_cluster_name>)
cd('/Server/ODI_server2')
set('ListenPort', int("<node2_odi_port>"))
updateDomain()
closeDomain()
```

2. Start all components using the `$DOMAIN_HOME/bitools/bin/start.sh` command.
3. Verify that all components are up and running using the `$DOMAIN_HOME/bitools/bin/status.sh` command.
4. Use Fusion Middleware Control to view system component availability. See [Use Fusion Middleware Control to View System Component Availability](#).
5. Use the Administration Console to view managed server availability. See [Use the Administration Console to View Managed Server Availability](#).

Configuring Oracle HTTP Server for High Availability

Use these steps to configure Oracle HTTP server (OHS) for high availability.

Install and configure the latest available supported version of Oracle HTTP Server on a separate machine. After installing and configuring OHS, configure the Oracle BI Applications URL in OHS.

1. Stop OHS using the command:

```
cd /u02/installation/OHS/Oracle_Home/user_projects/domains/
ohs_domain/bin ./stopComponent.sh ohs1
```

2. Add URL mappings for `odiconsole`, `oraclediagent`, `analytics` and `biacm` using this command:

```
cd /u02/installation/OHS/Oracle_Home/user_projects/domains/ohs_domain/
config/fmwconfig/components/OHS/instances/ohs1
```

Add this content to the `mod-wl_ohs.conf` file:

```
#odi agent

<Location /oraclediagent>
WebLogicCluster <node1_host>:<odi1_port>,<node2_host>:<odi2_port>
SetHandler weblogic-handler
</Location>

# ODI Console
<Location /odiconsole>
WebLogicCluster <node1_host>:<odi1_port>,<node2_host>:<odi2_port>
SetHandler weblogic-handler
</Location>

# analytics
```

```

<Location /analytics>
WebLogicCluster
<node1_host>:<bi_server1_port>,<node2_host>:<bi_server2_port>
SetHandler weblogic-handler
</Location>

# biacm
<Location /biacm>
WebLogicCluster
<node1_host>:<bi_server1_port>,<node2_host>:<bi_server2_port>
SetHandler weblogic-handler
</Location>

#bi-security-login
<Location /bi-security-login>
WebLogicCluster
<node1_host>:<bi_server1_port>,<node2_host>:<bi_server2_port>
SetHandler weblogic-handler
</Location>

```

3. Start OHS using the command:

```
cd /u02/installation/OHS/Oracle_Home/user_projects/domains/ohs_domain/bin ./
startComponent.sh ohs1
```

4. Verify that all the URLs configured above are working fine. For example:

- <http://ohshost:7777/oraclediagent/>
- <http://ohshost:7777/odiconsole>
- <http://ohshost:7777/analytics>
- <http://ohshost:7777/biacm>

5. Bring down one set of Oracle Analytics Server and Oracle Data Integrator servers and check if the URLs are redirecting to the Oracle Analytics Server and Oracle Data Integrator instances that are running.

Performing Post-Configure Steps for High Availability

Perform these steps after configuring Oracle BI Applications for high availability.

1. Log in to the Admin console in node1 and perform the following steps:
 - a. Navigate to **Environment** and then **Servers**.
 - b. Click **Admin Server** and then click the General tab.
 - c. Click **Lock & Edit**.
 - d. Update **Listen Address** field with the admin server host.
 - e. If default port is other than the default listen port, then update the **Listen Port** field with the appropriate port number.
 - f. Click **Save** and then click **Activate Change**.
2. Open Oracle Data Integrator (ODI) Studio or ODI Console. Go to Topology, click **Physical Architecture**, click **Agents**, and then click **OracleDIAgent**.
3. In Agent, update the Oracle HTTP server (OHS) hostname and port.

4. Create a network file system that is available in both hosts, such as filer2. Ensure that path `/filer2/HAShared` is available in both hosts.
5. Give write permission to the location to all, for example, `chmod -R 777 /filer2/HAShared`.
6. In node2, go to `<ORACLE_HOME>/bi/biapps/etl` and unzip `biapps_etl.zip` in the same location.
7. Edit the following parameters in the `BIAPPSConfig.properties` file available at `<DOMAIN_HOME>/config/fmwconfig/biinstances/coreapplication/BIAPPSConfig.properties` in both nodes:
 - `ODI_SERVER_PORT=<OHS_PORT>`
 - `ODI_SERVER_HOST=<OHS_HOST>`
8. Complete the functional setup and register the source in BIACM.
9. After registering the source in BIACM, note the data source number of that source (DSN). In the following steps, examples are given for DSN 215 and `SHARED_LOCATION` refers to `/filer2/HAShared`.
10. Once the source is registered, compare the folders in node2 with those of node1 and create the missing folders in node2. These folders may be missing in node2:
 - a. `<DOMAIN_HOME>/biapps/logs/etl`
 - b. `<DOMAIN_HOME>/biapps/logs/healthcheck/log`
 - c. `<DOMAIN_HOME>/biapps/logs/process`
 - d. `<DOMAIN_HOME>/biapps/shared/healthcheck/report`
11. Update the location of the source files for the physical schema of `BIAPPS_DW_FILE` data server in ODI Studio:
 - a. Log into ODI Studio. Navigate to Topology Manager, click **Technology**, and then click **File**.
 - b. Expand the `BIAPPS_DW_FILE` data server and double-click to open the physical schema.
 - c. In the Definition pane, for the Directory (Schema) and Directory (Work Schema) properties, specify the shared location where you copied the source files. Update the shared location in ODI Studio for the following by clicking **Topology**, then **Technologies**, and then **File** for:
 - `BIAPPS_DW_FILE`
 - `BIAPPS_DW_LOG_FILE`
 - `BIAPPS_DW_SHARED_FILE`
 - `EXT_STORAGE_FUSION_11_0_215`
 - `BIAPPS_DW_FILE` - Copy all the files/folders from `ORACLE_HOME/bi/biapps/etl/data_files/src_files/BIA_11` to `SHARED_LOCATION/BIA_11`.
 - `BIAPPS_DW_LOG_FILE` - Copy all the files/folders from `DOMAIN_HOME/biapps/logs` to `SHARED_LOCATION/logs`.
 - `BIAPPS_DW_SHARED_FILE` - Copy all the files/folders from `DOMAIN_HOME/biapps/shared` to `SHARED_LOCATION/shared`.
 - `EXT_STORAGE_FUSION_11_0_215` - Source registration may happen in any of the physical nodes. Hence, you must look into `DOMAIN_HOME` of

node1 and node2 to locate the folders mentioned in the following instructions:
Copy all the files/folders from DOMAIN_HOME/biapps/shared/215/ReplStg to SHARED_LOCATION/shared/215/ReplStg.

Copy all the files/folders from DOMAIN_HOME/biapps/shared/215/download to SHARED_LOCATION/shared/215/download.

Copy all the files/folders from DOMAIN_HOME/biapps/shared/215/srcFiles to SHARED_LOCATION/shared/215/srcFiles.

Copy all the files/folders from DOMAIN_HOME/biapps/shared/215/upload to SHARED_LOCATION/shared/215/upload.

Creating BI Schemas Using RCU with a RAC Database

Follow these steps for a successful configuration while creating Oracle Analytics Server schemas with the Repository Creation Utility (RCU) with a RAC database.

1. Install Oracle Fusion Middleware Infrastructure and Oracle Analytics Server as described in [Install Oracle Fusion Middleware Infrastructure \(WebLogic Server\)](#) and [Install Oracle Analytics Server](#).

Don't run Oracle Data Integrator and Oracle BI Applications yet.

2. Launch the RCU from `$ORACLE_HOME/oracle_common/bin/rcu` and select the **System Load and Product Load** option.
3. Provide the VIP or SCAN host of the RAC database with `sys` login details and click **Next**. Ensure that you select the Connection Parameters option and don't select the Connection String option.
4. Provide the prefix for the schemas, select all components, and then click **Next**.
5. View all the schemas to be created and click **Next**.
6. Verify the schema names, ensure that the schema password contains no special characters including the underscore, and click **Next** until completion.

You see all the schemas with data created in the database.

7. Verify that the `<Schema_Prefix>_STB.COMPONENT_SCHEMA_INFO` table is populated with applicable details.
8. Run the installers for Oracle Data Integrator and Oracle BI Applications and create the BIAPPS schema using the RCU.

Configure SSL

You can configure Oracle BI Applications with SSL for secured connections.

See My Oracle Support note Doc ID 2943320.1 for instructions and details.

Uninstalling Oracle BI Applications

Use the following steps to uninstall Oracle BI Applications.

1. Stop all running Oracle WebLogic services using this command: `$DOMAIN_HOME/bitools/bin/stop.sh`.
2. Drop all schemas by following these steps:

- a. Run the RCU by entering `./ORACLE_HOME/oracle_common/bin/rcu`.
- b. Select **Drop Schema** for each schema to drop and exit.
- c. Invoke RCU again using the command:

```
ORACLE_HOME/oracle_common/bin/rcu -compInfoXMLLocation
ORACLE_HOME/oracle_common/rcu/config/
ComponentInfo_biapps.xml -storageXMLLocation
ORACLE_HOME/oracle_common/rcu/config/Storage_biapps.xml
```

- d. Select **Drop Schema** and select all the BIAPPS schemas to drop.

 **Note:**

Ensure that you drop the schemas using the RCU; don't use any other SQL command.

3. Manually remove your Domain home directory.

For example, if your Domain home directory is `/u01/biapps/Middleware/user_projects`, then use the command

```
cd /u01/biapps/Middleware
rm -rf user_projects
```

.

4. Manually remove your Application home directory.

If there is another `user_project` directory within Oracle_home like `/u01/biapps/Middleware/Oracle_Home/user_projects`, then delete that using the following command:

```
cd /u01/biapps/Middleware/Oracle_Home
rm -rf user_projects
```

.

5. Remove the `domain_registry.xml` file in your Oracle home using the command `cd /scratch/user/installation/Middleware/Oracle_Home`
`rm -rf domain_registry.xml`.
6. Restart your system.
7. If you're trying to reconfigure, then keep `ORACLE_HOME` intact, create all schemas again, and then run the configure step.
8. If you're planning to complete the uninstallation process, then you need to remove the Oracle home directory and inventory folder that you provided while installing Oracle BI Applications.

4

Performing Post-Installation System Setup Tasks

Perform these post-installation system setup tasks for Oracle Business Intelligence Applications (Oracle BI Applications).

Topics:

- [Setting the Business Analytics Warehouse Connection in Oracle Data Integrator](#)
- [Performing Setup Steps for On-Premises Fusion Applications Sources](#)
- [Registering Source Systems and Propagating Connection Details to Oracle Data Integrator](#)
- [Enabling Offerings for Deployment](#)
- [Setting Languages for Data Load into the Business Analytics Warehouse](#)
- [Running the Domains Load Plan](#)
- [Mapping Externally Conformed Domains](#)
- [Granting Users Access to Configuration Manager FSM and Oracle Data Integrator](#)
- [Editing Preferred Currency Display Names and Enabling Document Currency](#)
- [Enabling Initialization Blocks](#)
- [Trimming the Repository](#)
- [Setting Up Fusion Applications Cloud Data Sources](#)
- [Setting Up Oracle Taleo Cloud Data Source](#)
- [Next Steps](#)

Setting the Business Analytics Warehouse Connection in Oracle Data Integrator

Use Oracle Data Integrator (ODI) Studio to set the connection properties in the ODI Repository of the physical schema associated with the BIAPPS_DW_FILE physical server.

To set the Oracle Business Analytics Warehouse physical schema connection in ODI:

1. Log in to ODI Studio using the Oracle BI Applications Administrator credentials.
2. In the navigation pane, go to **Topology**, and expand **Technologies**, and then expand **File**.
3. Double-click **BIAPPS_DW_FILE**.
4. In the Definition pane, for the Directory (Schema) and Directory (Work Schema) properties, specify the directory where you copied the source files to and include the sub-folders `data_files/src_files/BIA_11`.

For example:

`SOURCE_FILE_HOME/biapps/etl/data_files/src_files/BIA_11`

Performing Setup Steps for On-Premises Oracle Fusion Applications Sources

These procedures apply to on-premises Oracle Fusion Applications sources.

- [Creating a User for ETL](#)
- [Configuring the Oracle BI Repository for ETL](#)

Creating a User for ETL

The ETL process must be run by a user with appropriate data security privileges granted on the Oracle Fusion Applications tables from which data is extracted into Oracle Business Analytics Warehouse. For this purpose, the enterprise role named `FUSION_APPS_OBIA_BIEE_APPID` is provisioned during the installation of Oracle Fusion Applications with the appropriate ETL security privileges.

This procedure applies only to on-premises Oracle Fusion Applications sources. Work with your security administrator for Oracle Fusion Applications to complete these steps.

1. In the Oracle Fusion Applications LDAP system, create a new user.
For example, you might create a new user named `OBIA_ETL_USER`.
2. Make the user a member of the Enterprise Role `FUSION_APPS_OBIA_BIEE_APPID`.
3. Make a note of the user credentials.
4. When using the embedded LDAP for Oracle BI Applications, create a user with exactly the same credentials as the ETL user created in Oracle Fusion Applications LDAP.
Grant this `OBIA_ETL_USER` the `BIAdministrator Duty Role`.

Configuring the Oracle BI Repository for ETL

This procedure applies only to on-premises Oracle Fusion Applications sources.

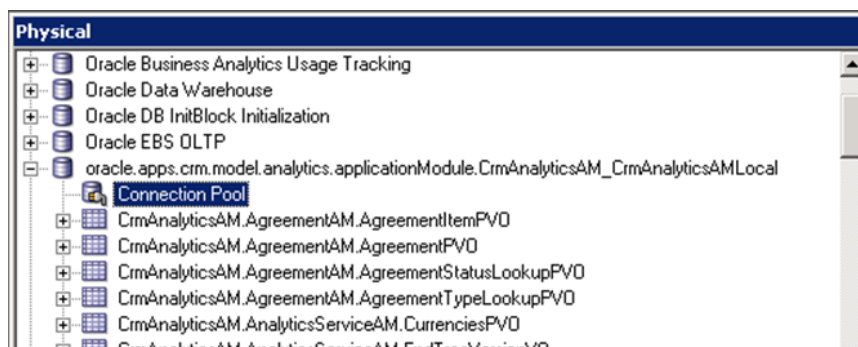
To configure the repository for ETL against on-premises Oracle Fusion Applications sources:

1. Log in to Oracle BI Administration Tool.
2. Open the Oracle BI Repository for Oracle BI Applications in offline mode.
3. In the Physical layer, configure a connection to the BIEE broker for each of the Oracle Fusion Applications pillar domains. Set the connection for each of these Physical layer objects:

Pillar	Physical Layer Object
CRM	<code>oracle.apps.crm.model.analytics.applicationModule.CrmAnalyticsAM_CrmAnalyticsAMLocal</code>

Pillar	Physical Layer Object
FSCM	oracle.apps.fscm.model.analytics.applicationModule.FscmTopModelIAM_FscmTopModelAMLocal
HCM	oracle.apps.hcm.model.analytics.applicationModule.HcmTopModelAnalyticsGlobalAM_HcmTopModelGlobalAMLocal

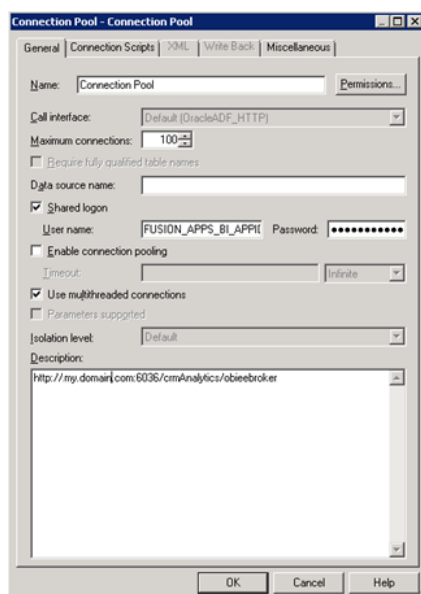
- a. Right-click one of the pillar objects, and select **Properties**.
- b. In the General tab of the Database dialog, select **Allow direct database requests by default**.
- c. Expand the pillar object, and double-click **Connection Pool**.



The Connection Pool dialog opens.

- d. In the General tab, enter `FUSION_APPS_BI_APPID` as the data source user name, and enter the password for this user.

You can obtain the password for the `FUSION_APPS_BI_APPID` user from your Oracle Fusion Applications administrator.



- e. In the Miscellaneous tab, in the Application Server URL field, enter the URL for the BIEE broker for the appropriate pillar, in the format:

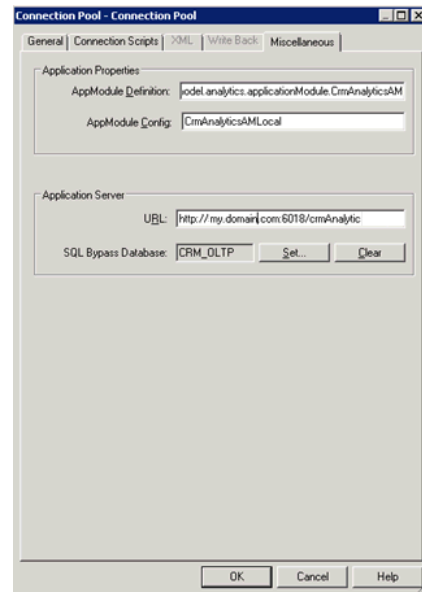
Http://<host>.<domain>:<port>/<pillar>Analytics/obieebroker

For example:

Http://hostName.myDomain.com:30615/crmAnalytics/obieebroker

- f. In the SQL Bypass Database field, enter <PILLAR>_OLTP.

For example: CRM_OLTP.



- g. Repeat steps 3a through 3f for each of the three pillars.

4. Set the value for these DSN variables.

These variables specify the connection to the Fusion Applications transactional database.

- CRM_OLTP_DSN
- FSCM_OLTP_DSN
- FBI_OLTP_DSN
- HCM_OLTP_DSN
- OLTP_DSN (used by the Marketing OLTP data source)

- a. In the menu bar, select **Manage**, then **Variables**.

- b. In the Variable Manager dialog, scroll down to locate the <PILLAR>_OLTP_DSN variable.

- c. Double-click the <PILLAR>_OLTP_DSN variable.

The Static Repository Variable dialog opens.

- d. In the **Default Initializer** text box, enter the connection information to the Fusion Applications transactional database in the tnsnames.ora format.

For example:

```
' (DESCRIPTION= (ADDRESS= (PROTOCOL=TCP)
(HOST=db_host_name.domain.com)
(PORT=<port>)) (CONNECT_DATA =
(SERVICE_NAME=<service_name>))) '
```

- e. Repeat steps 4a through 4d for each DSN.
5. Set the value for these user variables.
These variables specify the Oracle Fusion Applications transactional database user.
 - CRM_OLTP_USER
 - FSCM_OLTP_USER
 - FBI_OLTP_USER
 - HCM_OLTP_USER
 - OLTP_USER (used by the Marketing OLTP data source)
 - a. Obtain the user name for the users listed above from the repository for Oracle Transactional Business Intelligence.
 - b. Obtain the password for the users from the Oracle Fusion Applications administrator.
You'll need this password in a later step.
 - c. In the menu bar, select **Manage**, then **Variables**.
 - d. In the menu bar of the Variable Manager dialog, select **Action, New, Repository**, then **Variable**.
 - e. In the Variable Manager dialog, scroll down to locate the <PILLAR>_OLTP_USER and OLTP_USER variables.
 - f. Double-click the variable.
The Static Repository Variable dialog opens.
 - g. In the Default Initializer text box, enter the user name.
 - h. Repeat steps 4a through 4g for each user.
 6. Configure the connection pool for each of the OLTP objects listed in the Physical layer:
 - CRM_OLTP
 - FSCM_OLTP
 - FBI_OLTP
 - HCM_OLTP
 - Marketing OLTP
 - a. Expand the object.
 - b. Double-click **Connection Pool**.
For Marketing OLTP, double-click **Fusion OLTP Connection Pool**.
The Connection Pool dialog opens.
 - c. In the Data source name field, enter VALUEOF(<PILLAR>_OLTP_DSN).
For example: VALUEOF (CRM_OLTP_DSN).
For Marketing OLTP, enter VALUEOF (OLTP_DSN)
 - d. In the User name field, enter VALUEOF(<PILLAR>_OLTP_USER).

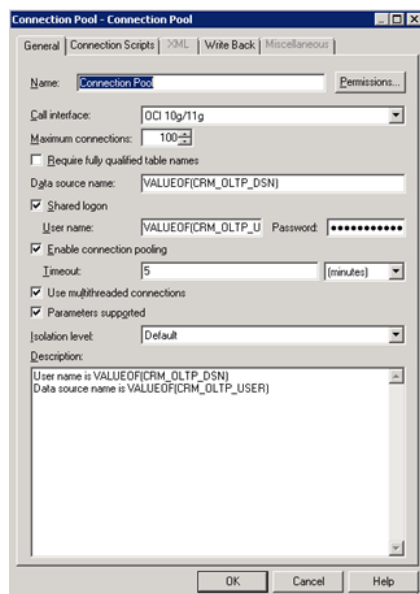
For example: `VALUEOF(CRM_OLTP_USER)`.

For Marketing OLTP enter `VALUEOF(OLTP_USER)`.

The user for CRM_OLTP, FSCM_OLTP, FBI_OLTP, and HCM_OLTP is the same. The user for Marketing OLTP is the value of OLTP_USER, which is different from the user for the other pillars.

- e. In the **Password** field, enter the password for the user.

Obtain this password from the Oracle Fusion Applications administrator.



- f. Repeat steps 6a through 6e for each of the OLTP objects.

Registering Source Systems and Propagating Connection Details to Oracle Data Integrator

Use this procedure to register a source system and propagate the connection details to ODI.

To register a source system:



Note:

If you want to register the Oracle Fusion Applications source, then follow the steps in [Register the Fusion Applications Source](#).

1. Launch Oracle BI Applications Configuration Manager using the URL `http://<host>:<port>/biacm`, and log in as the Oracle BI Applications Administrator user.
2. In the navigation pane, select the **Define Business Intelligence Applications Instance** link, which appears under System Setups.

The Source Systems tab is displayed.

3. Click the **Add** icon.

The Register Source dialog displays.

4. Register the source in Configuration Manager, specify these properties.

Property	Description
Product Line	Select the source instance product line. For Oracle Fusion sources, select Oracle Fusion.
Product Line Version	Select the version number of the source instance product line.
Source Instance Name	Specify a name for the source instance. This is the name given by the Oracle BI Applications System Administrator to a transactional system that serves as a source of data for Oracle BI Applications data load. The source instance is important because functional setup tasks, and the setting of values for objects such as parameters and domain maps are performed in relation to a source instance.
Description	(Optional) Enter additional information about the source instance.
Data Source Number	Enter a unique number to identify the source instance. The Data Source Number value specifies a data source so that the data can be identified in the Oracle Business Analytics Warehouse. Each source instance must have a unique Data Source Number.
Data Server	For all sources except PeopleSoft sources, a read-only list of data servers is displayed. For PeopleSoft Enterprise sources, in the Data Server Details window, select the appropriate data server for the source instance.

 **Note:**

Ensure that the Data Source Number value is the same in all environments (development, test, and production), otherwise the importing and exporting of data fails with error messages.

Click **Next**. The Register Source in ODI Topology page is active.

5. Register the source in ODI:

- a. Select **Global** from the Context drop-down list.

The Global value is required because the Oracle BI Applications interfaces are designed to use this value.

- b. In the Technology: Oracle tab, specify these properties:

Property	Description
ODI Data Server Name	Specify a name for the ODI Data Server that indicates the transactional OLTP database (source database). This ODI Data Server Name must be unique across all ODI data servers in the ODI Repository.

Property	Description
JDBC Driver	<p>The name of the JDBC driver for the transactional OLTP database (source database) connection.</p> <p>For Oracle Fusion sources, this property indicates the name of the JDBC driver for the Oracle BI Server. For example: <code>oracle.bi.jdbc.AnaJdbcDriver.</code></p> <p>Data from the Fusion Applications transactional system is retrieved from the Oracle BI Server.</p> <p>Oracle recommends that you use the default value.</p>
JDBC URL	<p>Specify the JDBC URL for the transactional OLTP database (source database) connection. The format for the URL is: <code>jdbc:oracle:thin:@<host>:<port>:<sid></code></p> <p>If your source is Oracle Fusion Applications on-premises, specify the JDBC URL for the Oracle BI Server. The format is <code>jdbc:oraclebi://<BI Server host>:<BI Server port>/ PrimaryCcs=<Cluster Controller Host>;PrimaryCcsPort=<Cluster Controller Port></code></p> <p>For example: <code>jdbc:oraclebi://biserverHost:9703/ PrimaryCcs=CCSHost;PrimaryCcsPort=9706</code></p>
Database Username	<p>Specify the database user name for the transactional OLTP database (source database).</p> <p>For Oracle Fusion sources, specify the user name of the ETL user. This is the user created in Creating a User for ETL.</p>
Database Password	<p>Specify the password for the transactional OLTP database (source database).</p> <p>For Oracle Fusion sources, specify the password for the ETL user.</p>
Test	<p>Use this button to test the connection to the transactional OLTP database (source database).</p>

- c. In the Technology: File System tab, specify these properties:

Property	Description
ODI Data Server Name	<p>Specify a name for the file system data server. This ODI Data Server Name must be unique across all ODI data servers in the ODI Repository.</p>
Host Name	<p>Specify the name of the host system where the file system (source files used for ETL) resides. This is the system to which you copied the source files.</p>
Path To File	<p>Specify the directory where you copied the source files to.</p> <p>The directory path must include the appropriate source-specific folder for your environment. The source-specific folders are located in <code>/biapps/etl/data_files/src_files</code>.</p> <p>For example, if your source system is Oracle EBS 12.1.1, the directory path would be: <code>SOURCE_FILE_HOME/biapps/etl/data_files/ src_files/EBS_12_1_1</code></p>

Property	Description
Array Fetch Size	The number of rows requested by ODI on each communication with the data server. This column may be hidden. To view this column, click View, Columns, Show All .
Batch Update Size	The number of rows in a single INSERT command. This column may be hidden. To view this column, click View, Columns, Show All .
JDBC Driver	Oracle recommends that you use the default value. This column may be hidden. To view this column, click View, Columns, Show All .
JDBC URL	Oracle recommends that you use the default value. This column may be hidden. To view this column, click View, Columns, Show All .

d. Click **Save and Close**.

This information is passed to ODI where the connections are created.

 **Note:**

The connection details aren't propagated to the ODI Repository until you enter and save the information on both the Register Source in the Configuration Manager page and the Register Source in ODI Topology page.

Enabling Offerings for Deployment

Use Configuration Manager to enable the Oracle BI Applications offerings that you've purchased and are deploying. The setup data relating to offerings is made visible in Configuration Manager when you enable the offering.

By default, if you've multiple source instances and you enable an offering, the offering is enabled for all source instances. If an offering won't source from all source instances, then you can disable the source instances from which the offering won't source. For example, suppose you've two source instances, Oracle EBS and PeopleSoft, and you're deploying Financial Analytics. If Financial Analytics sources only from PeopleSoft, then you'd disable the Oracle EBS source instance for the Financial Analytics offering.

To enable Oracle BI Applications offerings for deployment:

1. Log in to Configuration Manager as the Oracle BI Applications Administrator user.
2. Select the **Manage Business Intelligence Applications** link to display the Manage Business Intelligence Applications dialog.
3. Select the **Business Intelligence Application Offerings** tab.
4. Select **Enabled** next to each offering you want to deploy.

Enabling an offering makes the setup data associated with that offering available in Configuration Manager.

5. Click **Save** to save your changes.
6. To exit this dialog, click **Done**.

Disabling a Source Instance

You can disable a source instance from which an offering won't source.

1. Go to the **Business Intelligence Application Offerings and Associated Sources** tab.
2. Expand the appropriate offering.
3. Deselect **Enabled** for the source you want to disable.
4. Click **Save**.

Setting Languages for Data Load into the Business Analytics Warehouse

Oracle BI Applications supports data loads in multiple languages. You specify the languages for data loads in Configuration Manager.

Note:

PSFT and JDE customers must also configure the `LANGUAGE_BASE_OLTP` variable before executing the Domain-Only load plan. This is the base language of the PeopleSoft or JD Edwards OLTP and not necessarily the BI Apps base language. The `LANGUAGE_BASE_OLTP` variable doesn't apply to any other source system.

1. Log in to Configuration Manager as a user with the Oracle BI Applications Administrator duty role.
2. Select the **Manage Warehouse Languages** link to display the Manage Warehouse Languages page.
3. In the Manage Business Analytics Warehouse Languages tab, specify the languages from which data is to be extracted from the list of languages displayed in the table by selecting **Yes** from the Installed menu.

Note that American English is the default installed language. All other languages are disabled.

Note:

When you specify a language as being Installed, the Data Load Parameter `LANGUAGE_LIST` is populated internally with the list of selected languages. This parameter list is used during data extraction to extract data in the selected languages.

4. Click **Save** to save your changes.

Running the Domains Load Plan

You must define, generate, and run a domains load plan to load source-specific data into Configuration Manager tables. This enables Configuration Manager to display the appropriate source-specific values as choices in drop-down lists for setup objects.

Before you perform this step you must have completed the procedure [Registering Source Systems and Propagating Connection Details to Oracle Data Integrator](#), including registering the source in the Register Source in ODI Topology page. Additionally, if you've Fusion Cloud as a source in a load plan, then don't run such plans until you've set up the Oracle Fusion Applications cloud data source. See [Setting up Fusion Applications Cloud Data Sources](#).

To define, generate, and run the domains load plan:

1. Define the domains load plan:
 - a. In the Tasks pane of Configuration Manager, select **Manage Load Plans**, which appears under the Load Plans Administration heading.

The Manage Load Plans page is displayed.

- b. In the Load Plans toolbar, click the **Add** icon.

The Create Load Plan page is displayed.

- c. On the first page of the Create Load Plan series, specify this information:

Field	Description
Name	Enter a unique name for the load plan.
Description	(Optional) Enter additional information about the load plan.
Load Plan Type	Select Domain-only Extract and Load (SDE and SIL) .
Source Instances	Select a source instance. If you don't select an instance, then all instances are selected by default.

- d. Click **Next**.

The second page of the Create Load Plan series displays.

- e. In the Available Fact Groups tab, select the fact groups you want to include in the domains load plan definition.

The fact groups may belong to a hierarchy of fact groups. You can select only the top level parent fact group and not a child fact group.

A load plan must contain at least one fact group. Multiple fact groups may be selected from one or more data sources.

- f. Click **Save**.

A submenu is displayed with these options:

- Click **Save** to save the load plan. After a load plan is saved, it's displayed in the Load Plans master list.
- Click **Generate** to save and generate the load plan.

The generation process in Configuration Manager propagates the load plan properties to the ODI Repository, where the load plan is built.

2. If you didn't generate the load plan upon saving in the previous step, then generate it:

- a. In the Load Plans master list, select the domains load plan you defined in step 1.
- b. In the Load Plans toolbar, click the **Generate** icon.

The Generation Status column indicates the progress of the generation process. The process must complete successfully before you can move on to the next step. Click the **Refresh** icon to refresh the display.

3. Run the domains load plan by selecting it in the master list, and clicking the **Execute** icon on the toolbar.

Mapping Externally Conformed Domains

After you run the domains load plan, you must complete the mapping of externally conformed domains to point to a particular source (even if you've only one source) in Configuration Manager.

You must perform the mapping configuration for each domain listed in the Manage Externally Conformed Domains dialog.

User Access to Configuration Manager, FSM, and Oracle Data Integrator

On installation the Oracle BI Applications system is configured to use WebLogic Server embedded LDAP for authentication. Work with your security administrator to grant users access to Configuration Manager, Functional Setup Manager (FSM), and ODI.

Access to Configuration Manager and FSM is controlled through these duty roles:

- BI Applications Administrator Duty
- BI Applications Functional Developer Duty
- BI Applications Implementation Manager Duty
- Load Plan Operator Duty
- Load Plan Administrator Duty

The security administrator must grant the appropriate duty roles to a user based on the user's job responsibilities. See *Duty Roles for Access to Functional Setup Manager or Configuration Manager* in *Oracle Business Intelligence Applications Security Guide*.

The Oracle BI Applications administrator, load plan operator, and load plan administrator users require appropriate access to ODI. In addition to these users being created in the LDAP system, you must create these users in the ODI Repository and grant them the Supervisor profile or an appropriate ODI profile. The Oracle BI Applications administrator must be granted the Supervisor role in ODI. Work with your security administrator to grant the duty roles.

To manage security in ODI, see *Managing Security in Oracle Data Integrator* in *Administering Oracle Data Integrator*.

Additionally, you must complete the following steps to configure the **Signout** link to work for Configuration Manager on an Oracle Access Manager (OAM) enabled environment:

1. Execute the commands through WLST as the following:

```
addOAMSSOProvider(loginuri="/${app.context}/adfAuthentication",
logouturi="/oamssso/logout.html", autologinuri="/obrar.cgi")
Example:
cd <MW_HOME>/oracle_common/common/bin
./wlst.sh
connect ('biadmin','welcome1','t3://<WLS_HOST>:<WLS_PORT>');
wls:/>addOAMSSOProvider(loginuri="/${app.context}/adfAuthentication",
logouturi="/oamssso/logout.html", autologinuri="/obrar.cgi")
```

2. Stop and restart bi_server1.

Editing Preferred Currency Display Names and Enabling Document Currency

Oracle Analytics Server is installed with a set of preferred currencies with pre-configured preferred currency names and preferred currency codes. Preferred currency names are used on Oracle Analytics Server dashboards in the Currency drop-down on the My Account dialog\Preferences tab for a user logged into Oracle Analytics Server.

You can use the Manage Preferred Currencies dialog to edit the default currency display names. You edit preferred currency name values to change the currency labels that are displayed in all modules associated with BI dashboards. For example, you might want to change the "Local Currency" label from "Ledger Currency" to "Local Currency".

To edit currency display names:

1. Log in to Configuration Manager as the Oracle BI Applications Administrator user.
2. Under System Setups, select **Manage Preferred Currencies** to display the Manage Preferred Currencies dialog.
3. Select a currency in the Preferred Currencies list.
Selecting the currency displays the associated modules in the bottom table.
4. Click the value in the Preferred Currency Name column (or click the Edit icon) to display the Preferred Currency Name dialog.
5. In the Preferred Currency Name field, specify a currency name.
This is the name that appears for end users in the Currency drop-down list, located in Preferences tab of the My Accounts dialog of Oracle Analytics Server.
6. Click **Save and Close**.

Enabling Document Currency

By default, document currency is excluded from the Currency drop-down list (located in the Preferences tab of the My Accounts dialog in Oracle Analytics Server).

To include Document currency in the Currency drop-down list, you must remove a specific filter from all of these security groups in the repository:

- OBIA_PROJECT_CURRENCY_PREFERENCES
- OBIA_HCM_CURRENCY_PREFERENCES

- OBIA_SCM_CURRENCY_PREFERENCES
- OBIA_FINANCIAL_CURRENCY_PREFERENCES
- OBIA_MFG_CURRENCY_PREFERENCES
- OBIA_PROCUREMENT_CURRENCY_PREFERENCES
- OBIA_MARKETING_CURRENCY_PREFERENCES
- OBIA_PARTNER_CURRENCY_PREFERENCES
- OBIA_CRM_CURRENCY_PREFERENCES
- OBIA_SALES_CURRENCY_PREFERENCES



Note:

The security groups listed above are displayed in Oracle BI Administration Tool on the Application Roles tab on the Identity Manager dialog.

1. In Oracle BI Administration Tool, click **Manage**, then **Identity**, to display the Identity Manager dialog.
2. In the Identity Manager dialog, click **BI Repository**.
3. Display the Application Roles tab.
4. For each of the security groups in the repository list:
 - a. Double-click the security group to display the Application Role - <Name> dialog.
 - b. Click **Permissions**, and then display the Data Filters tab.
 - c. In the Data Filter column, remove the filter:

```
AND "Core"."Fact - Preference List"."Currency Preference Code"  
<> 'Document Currency'
```

For example, the filter before editing:

```
"Core"."Fact - Preference List"."Module Code" = 'PROJECT_AN' AND  
"Core"."Fact - Preference List"."Currency Preference Flag" in  
( 'W', 'B') AND "Core"."Fact - Preference List"."Currency  
Preference Code" <> 'Document Currency'
```

For example, the filter after editing:

```
"Core"."Fact - Preference List"."Module Code" = 'PROJECT_AN' AND  
"Core"."Fact - Preference List"."Currency Preference Flag" in  
( 'W', 'B')
```

5. Save the changes.

If you edited the metadata repository in Offline mode, then you must use Oracle Enterprise Manager Fusion Middleware Control to upload the repository.

**Tip:**

To upload metadata to Oracle Analytics Server, see Upload Repository Command.

Enabling Initialization Blocks

For all source systems other than Oracle Fusion Applications, you must enable initialization blocks. Initialization blocks are enabled for Oracle Fusion Applications by default.

This procedure uses forward slashes to separate directories, indicating the UNIX/Linux directory structure. For Windows, substitute a backslash; the directory structure is the same as that for UNIX/Linux.

1. Copy the sample `config.txt` file from `ORACLE_HOME/bi/biapps/biarpdutil/bin` to `ORACLE_HOME/bi/biapps/biarpdutil/lib`.
2. Download the Oracle BI Repository file for Oracle BI Applications to the library location using this command:

```
<DOMAIN_HOME>/bitools/bin ./datamodel.sh downloadrpd -O
<export_location> -SI <service_instancename > -U <username> -P <password>
```

For example,

```
./datamodel.sh downloadrpd -O <ORACLE_HOME>/bi/biapps/biarpdutil/lib/
OracleBIApps.rpd -SI biainstance -U username -P password
```

3. Add `<DOMAIN_HOME>/bitools/bin` to the `PATH` system variable.
4. Edit the `config.txt` file that you copied to `ORACLE_HOME/bi/biapps/biarpdutil/lib`:

```
RPD File Name <RPD file name>
RPD Password <RPD password>
RPD Database List <Comma separated list of database names in RPD physical layer
for which the initialization block has to be enabled>
```

The values in the repository database list must include all initialization blocks related to the source system that need to be enabled. Oracle BI Server must be included as a value in the repository database list if an initialization block that needs to be refreshed from the Oracle Analytics Server server is to be enabled.

5. Go to the location `<DOMAIN_HOME>/bitools/bin`, make a copy of `biserverxmlgen.sh`, rename it to `biserverxmlgen`, and give the new file `biserverxmlgen` all the required permissions.
6. Execute the `ManageInitBlock.jar` utility, located in `ORACLE_HOME/bi/biapps/biarpdutil/lib`, using this command:

```
java -classpath ManageInitBlock.jar oracle.apps.biarpdutil.ManageInitBlock >log.txt
```

An output repository is generated. The file is named `Output_RPD.rpd`. All relevant initialization blocks are enabled in `Output_RPD.rpd`.

7. Deploy the `Output_RPD.rpd` to Oracle Analytics Server.

To deploy an Oracle BI Repository, see *Administering Oracle Analytics Server*.

Trimming the Repository

You can trim the repository so that it includes only the projects that are relevant to your deployment.

Oracle BI Applications delivers a full repository with projects for all the BI Applications modules. This full repository is deployed to the Oracle BI Server. Although optional, trimming the repository makes the Oracle BI Server startup process faster and also makes patching quicker.

Note:

You mustn't trim the repository for on-premises Fusion Applications sources. The full repository is required for ETL in these environments.

The steps for trimming the repository depend on the status of your deployment:

- If the repository hasn't been customized for your deployment: Extract the projects for the products that your organization has purchased. You don't need to perform a merge. See [Extracting Projects From Full Repository](#).
- If the repository has been customized for your deployment: Extract the applicable projects from the full (delivered) repository, and, additionally, merge that repository with your customized repository. See [Extracting Projects From Full Repository](#) and [Merging Repositories](#).

Extracting Projects From Full Repository

Use this procedure to extract projects from the full repository. The end result of this process is a trimmed repository.

To extract from the repository the projects for the products you've purchased, you can use the `extractprojects` utility only with binary repositories in RPD format. This utility is available at location `ORACLE_HOME/user_projects/domains/bi/bitools/bin`.

1. For extracting a single project, use this example:

```
extractprojects -B input_rpd -O output_rpd -I project1 Provide your
password for the repository. This example extracts project1 from my_repos.rpd
and creates a new repository called extract_repos.rpd.
```

2. For extracting multiple projects, use this example:

```
extractprojects -B input_rpd -O output_rpd -I "project_name1" -I
"project_name2"-I "project_name3" (and so on) where
input_rpd is the name and path of the full (delivered) repository and from which
you want to extract the project or projects (for example, OracleBIApps.rpd).
output_rpd is the name and path of the repository that you want to create with the
extracted projects (for example, OracleBIAppsTrimmed.rpd).
project_name is the name of the repository project that you want to extract.
You're prompted to enter the encryption password for the repository (input_rpd).
```

The list of projects in the repository includes:

- Financial Analytics Fusion Edition
 - Human Resources Analytics Fusion Edition
 - Marketing Analytics Fusion Edition
 - Partner Analytics Fusion Edition
 - Project Analytics Fusion Edition
 - Sales Analytics Fusion Edition
 - Supply Chain and Order Management Analytics Fusion Edition
 - Student Information Analytics
 - Service Analytics
 - Price Analytics
 - Manufacturing Analytics
3. Save and rename the trimmed repository. Make sure the name identifies this repository as one that has been trimmed, for example, `OracleBIAppsTrimmed.rpd`.

Merging Repositories

Use this procedure only if you've customized the repository and are subsequently trimming the repository.

1. In the Oracle BI Administration Tool, open in offline mode the trimmed Oracle BI Repository that you created in the procedure [Extracting Projects From Full Repository](#), for example, `OracleBIAppsTrimmed.rpd`.
2. On the menu bar, click **File**, and then click **Merge**.
3. In the Select Original Repository dialog box, select the repository `OracleBIApps..`, which is the full repository.
4. Enter the password for the original repository, and then click **OK**.
5. Click **Select** for the Modified Repository field.
The Select Modified Repository dialog opens.
6. Select the repository that contains the customizations you made to the repository file, for example, `OracleBIAppsCustom..`
7. Click **Open**, and enter the password for the previously customized repository, and then click **OK**.
8. In the Decision drop-down list, select the action you want to take regarding the repository change, or accept the default action.
9. To locate subsequent rows with empty Decision fields, click the **Decision** header cell.
After all rows have a value in the Decision field, the **Merge** button becomes enabled.
10. Click **Merge**.
A message appears after the merge concludes successfully.
11. On the menu bar, click **File**, and then click **Save As**.
Save the current repository using a new name, such as `OracleBIAppsFinal.rpd`.

Next Steps After Setup — Functional Configuration

After you complete the setup tasks, Oracle BI Applications is ready for functional configuration, which is accomplished using Configuration Manager and FSM.

 **Note:**

If you're deploying data from Oracle Fusion Applications Cloud source system, complete the tasks in [Setting Up Fusion Applications Cloud Data Sources](#) before performing functional configuration.

To enable functional configuration, you must provide the Configuration Manager URL to your implementation managers and functional developers. To learn about the functional configuration process for Oracle BI Applications, see *Oracle Business Intelligence Applications Configuration Guide*.

In addition, for Oracle Fusion Applications sources, after Oracle BI Applications analyses and dashboards have been tested and the data validated, the Oracle BI Applications system administrator needs to inform the Oracle Fusion Applications Administrator to enable the appropriate embedded analyses and dashboard (ADR) regions in Oracle Fusion Applications. The ADR regions are enabled using FSM for Oracle Fusion Applications.

5

Setting Up Fusion Applications Cloud Data Sources

Follow these procedures to deploy Oracle Business Intelligence Applications (Oracle BI Applications) with a Fusion Applications Cloud data source.

Topics:

- [Overview of Oracle Fusion Applications Cloud Source System Support](#)
- [Fusion Applications Cloud Source System Deployment Options](#)
- [Fusion Applications Cloud Source System Deployment Roadmap](#)
- [Create a Service Request to Deploy Oracle BI Cloud Connector Console](#)
- [Provision a User For Export and Import of BI Data from Fusion UCM](#)
- [Provision a User for Oracle Business Intelligence Cloud Connector Access](#)
- [Register the Fusion Applications Source](#)
- [Create a Load Plan and Specify Offerings and Fact Groups to Load](#)
- [Extract Data into Universal Content Management using BI Cloud Connector Console](#)
- [Synchronize Deletes in Your Cloud Extract](#)
- [Configure Proxy Settings for WebLogic Server](#)
- [Managing Fusion Flex Extensions in Release 11.1.1.10.3](#)
- [Set Up Key Flex Fields for Financials Fusion Applications Source Data](#)
- [Setting Up Key Flex Fields for HCM Fusion Applications Source Data](#)

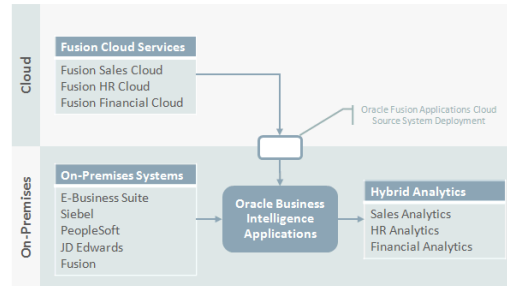
Overview of Oracle Fusion Applications Cloud Source System Support

You can enable your on-premises deployment of Oracle BI Applications to consume data from your Fusion Cloud Services, providing you with a comprehensive set of analytics that span your on-premises and cloud service systems to give you a complete view of your business.

Along with the many supported on-premises source business systems, support for Oracle Fusion Applications Cloud Services enables a robust hybrid analytics solution to deliver the complete picture regardless of where the data originates.

Oracle Business Intelligence Applications: Hybrid Analytics

- Oracle Fusion Applications Cloud Service Adapter for OBIA enables hybrid analytic scenarios.
- Enables Oracle Application Cloud Services to be OBIA Data Sources



You can securely connect your on-premises Oracle BI Applications deployment to your Oracle Fusion Applications Cloud Service, which automates the flow of data from your source system to your Oracle Business Analytics Warehouse, enabling your hybrid analytics solution.

Key features include:

- Extracts are from VOs and not direct from DB.
- Leverages Standard Security Methods of Source Cloud Systems to secure data in-transit.

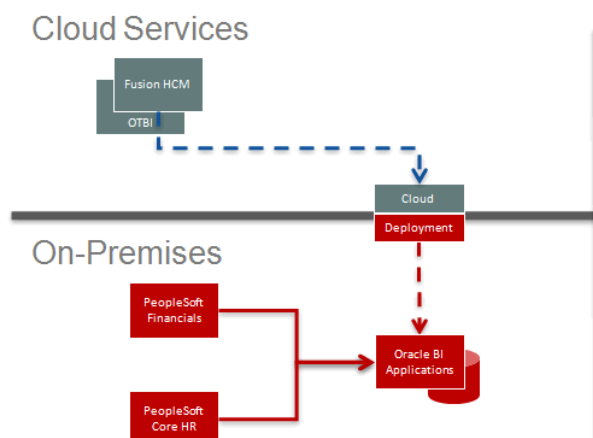
Fusion Applications Cloud Source System Deployment Options

When migrating to the Cloud, you can enable your on-premises deployment of Oracle BI Applications to consume data from your Fusion Cloud Services using these deployment options.

Hybrid Data Sources Category

Deployment Option: Hybrid Data Sources

Fusion SaaS & On-Premises PeopleSoft, Oracle BI Applications



Scenario:

Customer begins with the following on-premises applications:

- PeopleSoft Financials
- PeopleSoft Core HR
- Oracle BI Applications

Wants to migrate some HR functionality to Fusion HCM in the cloud, yet retain cross-functional analytics with both cloud and on-premises systems.

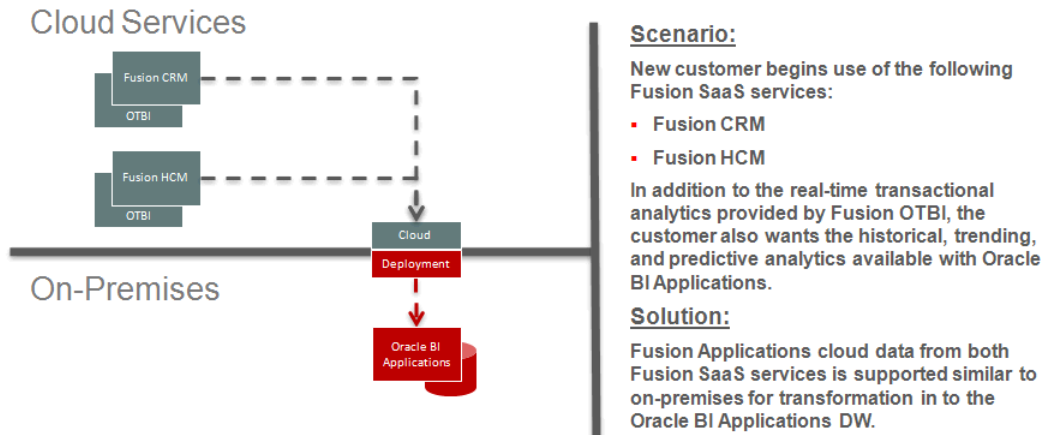
Solution:

Fusion Applications cloud data is supported similar to on-premises data sources in to the Oracle BI Applications DW.

Cloud-Only Data Sources

Deployment Option: Cloud-Only Data Sources

Fusion SaaS & On-Premises Oracle BI Applications



Fusion Application Cloud Source System Deployment Roadmap

You can deploy a Fusion Applications Cloud data source to your on-premises deployment of Oracle BI Applications.

Prerequisites for deployment:

- Oracle BI Applications 11.1.1.10.3
- Oracle Cloud Applications; refer to the Oracle BI Applications 11.1.1.10.3 Certification Matrix for the certified Cloud Application releases.

To deploy a Fusion Applications Cloud data source:

1. Create a service request to deploy Oracle BI Cloud Connector Console (BICCC) on your Fusion Applications pod. Additionally, create a service request to review sizing of your Fusion Applications pod for Oracle BI Cloud Connector Console usage. See [Create Service Requests for Oracle BI Cloud Connector Console](#).
2. Provision a new or existing Fusion Applications user with export and import privileges for Oracle Business Intelligence. See [Provision a User For Export and Import of BI Data from Fusion UCM](#).
3. [Provision a User for Oracle Business Intelligence Cloud Connector Access](#).
4. [Register the Fusion Applications Source](#).
5. [Create a Load Plan and Specify Offerings and Fact Groups to Load](#).
6. [Extract Data into Universal Content Management using BI Cloud Connector Console](#).
7. [Synchronize Deletes in Your Cloud Extract](#).
8. [Configure Proxy Settings for WebLogic Server](#). If you have already configured WebLogic server to allow any external connection, then you can skip this step.

9. Set up the Key Flex Fields before you execute the load plan (created in the step 5) if you are deploying Key Flex Fields for Financials. See [Setting Up Key Flex Fields for Financials Fusion Applications Source Data](#).
10. [Set Up Key Flex Fields for HCM Fusion Applications Source Data](#).

Creating Service Requests for Oracle Business Intelligence Cloud Connector Console

Create a service request if you're deploying Oracle Fusion Applications data sources.

Sign in to My Oracle Support (support.oracle.com) and create a service request to review sizing of your Oracle Fusion Applications pod for Oracle BI Cloud Connector Console usage. When the service request is received, the Oracle Cloud Services team reviews the available disk space and UCM tablespace on the applicable Oracle Fusion Applications pod and increases the space as required.



Note:

In Oracle Fusion Applications Release 10 and higher, Oracle BI Cloud Connector Console is automatically deployed and no separate service request is required to deploy it.

Log a service request, and specify the following details:

- Action: Review sizing for BICCC
- Problem type: Specify this information:
 - Service Type: **Oracle Fusion Global Human Resource Cloud Service**
 - Problem Type: **Hosting Services – Application**
 - Sub-Problem Type: **Review sizing for BICCC**
 - BIA Deployment: **On-premise**
 - User ID: <User ID for Business Intelligence Cloud Connector Console (BICCC) access>
 - FA POD details: <POD ID>
 - FA POD: <Stage and Production>
- Inputs: This service request initiates the review of available disk space and UCM tablespace on the applicable Fusion pod. Provide the Oracle Fusion Applications POD details in the service request. This must include the POD identifier (the URL used to access the Fusion environment).

When the service request is resolved, Oracle Cloud Services provides you with the UCM Server and Oracle BI Cloud Connector Console URLs.

Provisioning a User for Export and Import of BI Data from Fusion UCM

Provision a new or existing user in Oracle Fusion Applications with privileges to download data from Fusion Universal Content Management (UCM).

This task is a part of the [Fusion Applications Cloud Source System Deployment Roadmap](#).

To provision a user for Fusion access:

1. Create a new user, or use an existing user in Oracle Fusion Applications that has privileges to download data from Fusion UCM.
2. Using Oracle Authorization Policy Manager (APM), assign the following application role to the applicable new or existing user:

Application Role	Display Name
OBIA_EXTRACTT	Upload and download data from on-premises system to cloud system
RANSFORMLOA	
D_RWD	

Provisioning a User for Oracle BI Cloud Connector Console Access

Use these steps to provide additional users access to the Oracle BI Cloud Connector Console (BICCC) user interface.

The service request that you created to deploy the Oracle BI Cloud Connector Console performs the setup for the identified user. See [Creating Service Requests for Oracle Business Intelligence Cloud Connector Console](#). Complete these steps only if additional users need to access the console.

For Oracle Fusion Applications Data Sources

Use the Security Console in Oracle Fusion Applications to create an administrative role that inherits the Oracle BI Cloud Connector Console privileges from existing roles and assign the user to that role.

To provision a user:

1. In Oracle Fusion Applications, navigate to the **Security Console** in the Navigator.
2. In the Security console, create a BICC_ADMIN role.
 - a. Click **Create Role**.
 - b. In the Basic Information page, enter the following values and click **Next**.
 - i. **Role Name:** BICC_ADMIN
 - ii. **Role Code:** BICC_ADMIN
 - iii. **Role Category:** BI – Abstract Roles
 - c. Click the **Add** icon in the Role Hierarchy list.

- d. In the Add Role Membership dialog, search for `ESS`.
- e. In the search results, confirm that the ESS Administrator role is displayed and then click **Add Role Membership**.
- f. Search for `ORA_ASM_APPLICATION_IMPLEMENTATION_ADMIN_ABSTRACT` and click **Add Role Membership**.
- g. Search for `OBIA_EXTRACTTRANSFORMLOAD_RWD` and click **Add Role Membership**.
- h. Close the Add Role Membership dialog.
- i. Click **Next**.
- j. In the Users page, click **Add User**.
- k. In the Add User dialog, search for the name of the user you want to assign access to, and then click **Add User to Role**.
- l. Close the Add User dialog.
- m. Click **Next**.
- n. Click **Save** and close.

Registering the Fusion Applications Source

Register the Oracle Fusion Applications source.

Ensure that the applicable SDS schemas have been created during installation and configuration of Oracle BI Applications. For Oracle Fusion Applications cloud source systems, ensure that you select the Oracle Fusion 11 SDS schema in the Select Components dialog of Oracle Fusion Middleware Repository Creation Utility. See [Creating Schemas Using Business Analytics Applications Suite RCU](#)



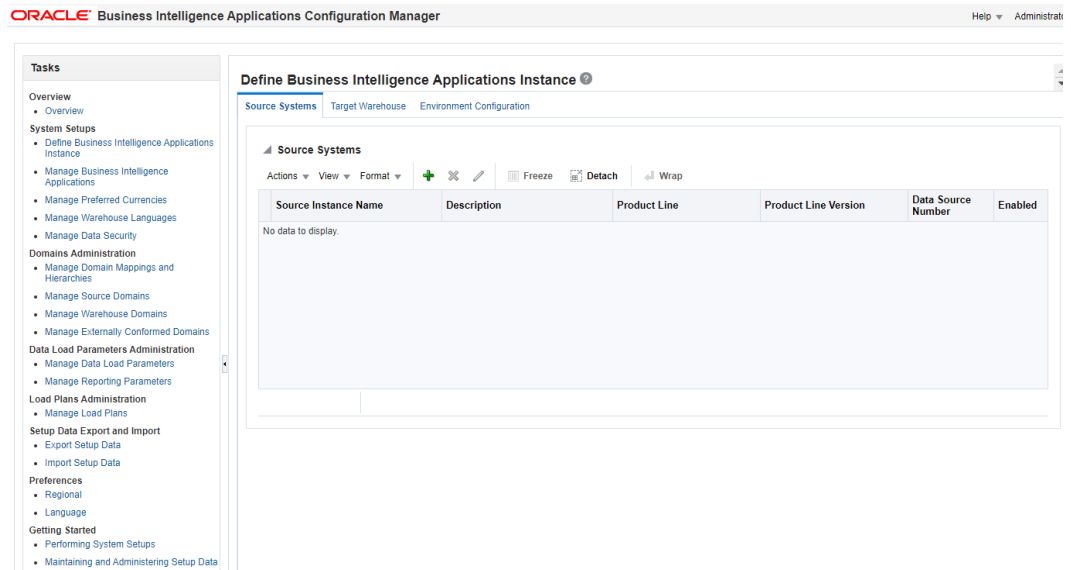
Note:

If you haven't created the applicable SDS schemas during installation and configuration of Oracle BI Applications, then you can create them as a post-installation step.

This task is part of the [Fusion Applications Cloud Source System Deployment Roadmap](#).

To Register the Oracle Fusion Applications Source:

1. Log in to Oracle BI Applications Configuration Manager as a user that has Enterprise Role `-ASM_APPLICATION_IMPLEMENTATION_ADMIN_ABSTRACT`.
2. In the Tasks pane under System Setups, select the **Define Business Intelligence Applications Instance** link to display the Source Systems tab.



3. Click the **Add** icon (+) to display the Register Source dialog.

Register Source

Source Details Connection Details

Back Next

* **Product Line**

* **Product Line Version**

* **Source Instance Name**

Description

* **Data Source Number**

Source Load Technology

4. In the Register Source in Configuration Manager page, specify this information:
 - In the Product Line drop-down list, select **Oracle Fusion**.
 - In the Product Line Version, select the Oracle Fusion version being used.
 - In the Source Instance Name field, specify a unique name to identify the source system.
 - Optionally, in the Description field, enter a short description to help administrators and implementers identify and use this data source in Configuration Manager.
 - In the Data Source Number field, enter an integer or use the spinner to specify a number to identify data in the Oracle Business Analytics Warehouse. This number must be unique within Configuration Manager.
 - In the Source Load Technology, select `ODI External File` to define the cloud adapter mode.

5. Click **Next** to display the external file location details.
6. In the External File Location Details dialog, specify the following details, and then click **Save**:

Register Source

The screenshot shows the 'Register Source' dialog with a progress indicator at the top. The 'External File Location Details' step is active. Below the progress bar are 'Back' and 'Next' buttons. The main content area is titled 'UCM Connection Configuration' and contains the following fields:

- * Protocol**: A dropdown menu with 'https' selected.
- * Host**: A text input field containing 'fus[redacted].com'.
- Port**: An empty text input field.
- * User Name**: A text input field containing 'faadmin'.
- * Password**: A text input field with masked characters '.....'.

Below these fields is a 'Test UCM Connection' button. At the bottom of the dialog, the 'Staging Folder Configuration' section is partially visible.

- a. Protocol – Select https.
- b. Host and Port – Specify the host and port details provided by the Oracle Cloud Services in your service request when it's closed.
- c. User Name and Password – Specify the user name and password of the user that you provisioned for export and import of BI data from Fusion Universal Content Management server. See [Provisioning a User for Export and Import of BI Data from Fusion UCM](#).
- d. Click **Test UCM Connection** to test the connection.

Creating a Load Plan and Specifying Offerings and Fact Groups to Load

Create a load plan and specify offerings and fact groups to load.

To create a load plan:

1. In the Tasks pane, click **Manage Load Plans** to display the Load Plans dialog.
2. Click **Add (+)** to display the Define Load Plan dialog.

Define Load Plan ?

The screenshot shows the 'Define Load Plan' dialog box. At the top, there is a progress bar with two steps: 'Enter Name and Description' (the first step is active, indicated by a blue circle) and 'Select Fact Groups'. To the right of the progress bar are 'Back' and 'Next' buttons. Below the progress bar, there are four input fields: a text field for '* Name', a text field for 'Description', a dropdown menu for 'Load Plan Type' (currently showing 'Source Extract and Load (SDE, SIL and PLP)'), and a dropdown menu for '* Source Instances' (currently showing 'Fusion Applications 13').

3. Use the Name field to specify a unique name, and in the Source Instances drop-down list select the Fusion source, then click **Next** to display the Select Fact Groups page.
4. Select applicable offerings and fact groups.

Define Load Plan FA load plan ?

Enter Name and Description | Select Fact Groups | Back | Next

Available Selections | Selected Fact Groups

Name	Code	Selected
Fusion Applications 13	Data Source	<input checked="" type="checkbox"/>
Oracle Custom Analytics		<input type="checkbox"/>
Oracle Customer Data Management Analy		<input type="checkbox"/>
Oracle Financial Analytics		<input type="checkbox"/>
Oracle Human Resources Analytics		<input type="checkbox"/>
Oracle Marketing Analytics		<input type="checkbox"/>
Oracle Partner Analytics		<input type="checkbox"/>
Oracle Procurement and Spend Analytics		<input type="checkbox"/>
Oracle Product Information Management A		<input type="checkbox"/>
Oracle Project Analytics		<input type="checkbox"/>

View ▼ Format ▼ | Freeze | Detach | Filter | Sort | Refresh | >>

Save | Generate | Cancel

5. Click **Save**.
6. In the list of Load Plans, select the new load plan, and click **Generate**.

Setting up ODI Agent Concurrent Sessions

For optimal performance, set the maximum number of sessions for Oracle Data Integrator (ODI) Agent to no more than five. To set up the maximum number of sessions supported by an agent:

1. Log into ODI Studio.
2. Display the **Topology** tab, and navigate to the **Physical Architecture** tab.
3. Expand **Agents** and select the applicable ODI Agent.

The screenshot shows the Oracle BI Cloud Connector Console interface for configuring an agent. The 'Agent' section includes the following fields:

- Name: OracleDIAgent
- Host: bl[redacted].com
- Port: 15001
- Web application context: oraclebiagent
- Protocol: http
- Maximum number of sessions: 5
- Maximum number of threads: 5
- Maximum threads per session: 0

The 'Session Blueprint Cache Management' section includes:

- Maximum cache entries: 100
- Unused Blueprint Lifetime (Sec): 500

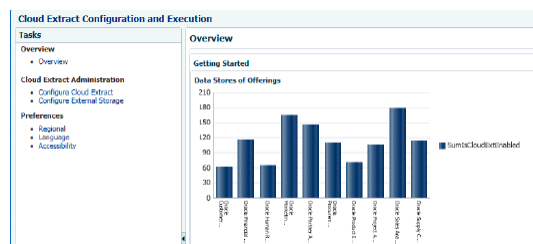
- In the agent details pane, set the maximum number of sessions and threads supported by the agent as 5. If you're creating a new agent, then ensure that you set the maximum number of sessions supported by the agent as 5.

Extracting Data into Universal Content Management using Oracle BI Cloud Connector Console

You can extract data from your cloud source into the Universal Content Management (UCM) server using the Oracle BI Cloud Connector Console (BICCC).

Use the Oracle BI Cloud Connector Console URL provided to you in the SR that you logged to deploy Oracle BI Cloud Connector Console and the login details of the applicable user to log into this console. See [Create a Service Request to Deploy Fusion Applications Cloud Source System Data](#), [Provision a User For Export and Import of BI Data from Fusion HCM](#), [Provision a User for Oracle Business Intelligence Cloud Connector Access](#)

You can also start Oracle BI Cloud Connector Console using a HTTP URL based on the following format: `http://<FA OHS Host>:<FA OHS Port>/biacm`



To extract data from a cloud source to UCM:

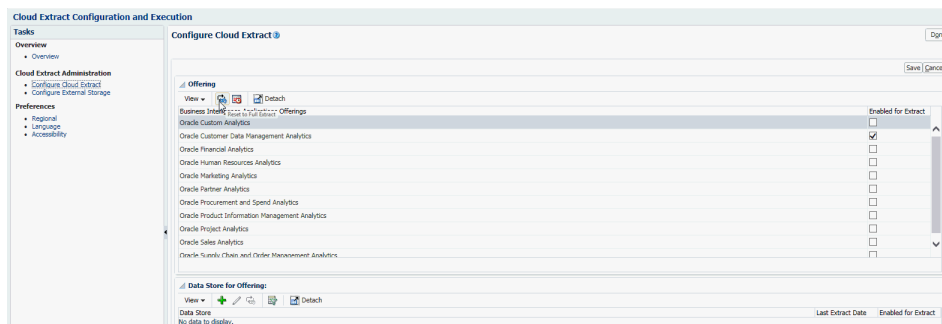
Note:

Oracle BI Applications on-premises users must extract data to UCM.

- From the landing page, select the **Configure Cloud Extract** link on the Tasks bar.
- In the list of Offerings, select the check box next to the Cloud source system functional areas that you want to deploy.



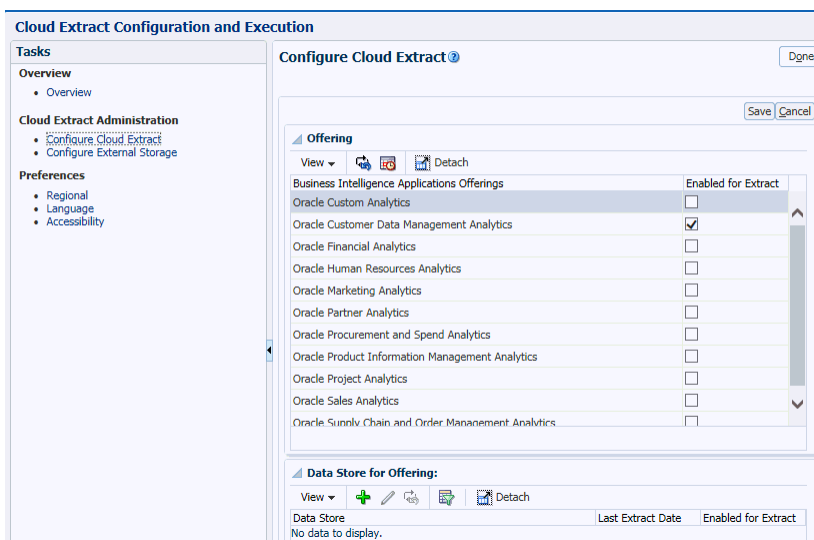
3. Click **Save**, then **Done**
4. To reset the last extract date so that a full data load is performed at the next load for the selected Offering, instead of an incremental load, click **Reset to Full Extract** icon in the tool bar and click **OK** when the application displays the warning message. You typically use this option if your business requirements have changed or if fact data has been corrupted. You can reset to full extract at offering and VO level.



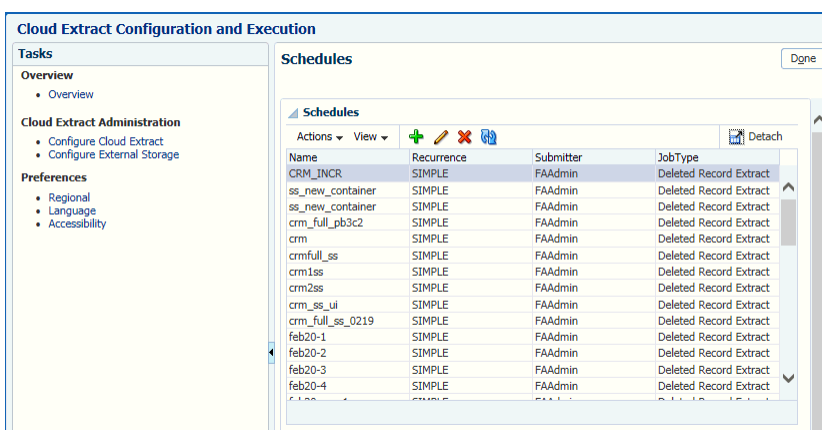
Note:

You set up a Scheduled Load Plan in Oracle BI Applications Configuration Manager (BIACM) to extract from UCM server into Oracle Business Analytics Warehouse. To reset the extraction on Oracle Business Analytics Warehouse, use the BIACM option: Manage Load Plans\Execute\Execute Reset Data Warehouse Scenario. See Manage Load Plans Toolbar Options in *Oracle Business Intelligence Applications ETL Guide*.

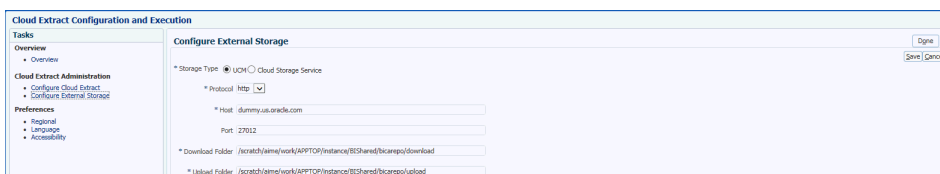
5. To schedule the extract, click the **Schedule** icon in the tool bar.



- Click the + icon to create a new schedule.



- Specify when you want to perform the cloud extract.
- Click **Schedule**, then **Done**.
- Click the **Configure External Storage** link on the Tasks bar.
- Select **UCM** as the storage type.




- View the configuration information displayed on this page and click **Save**, then **Done**.

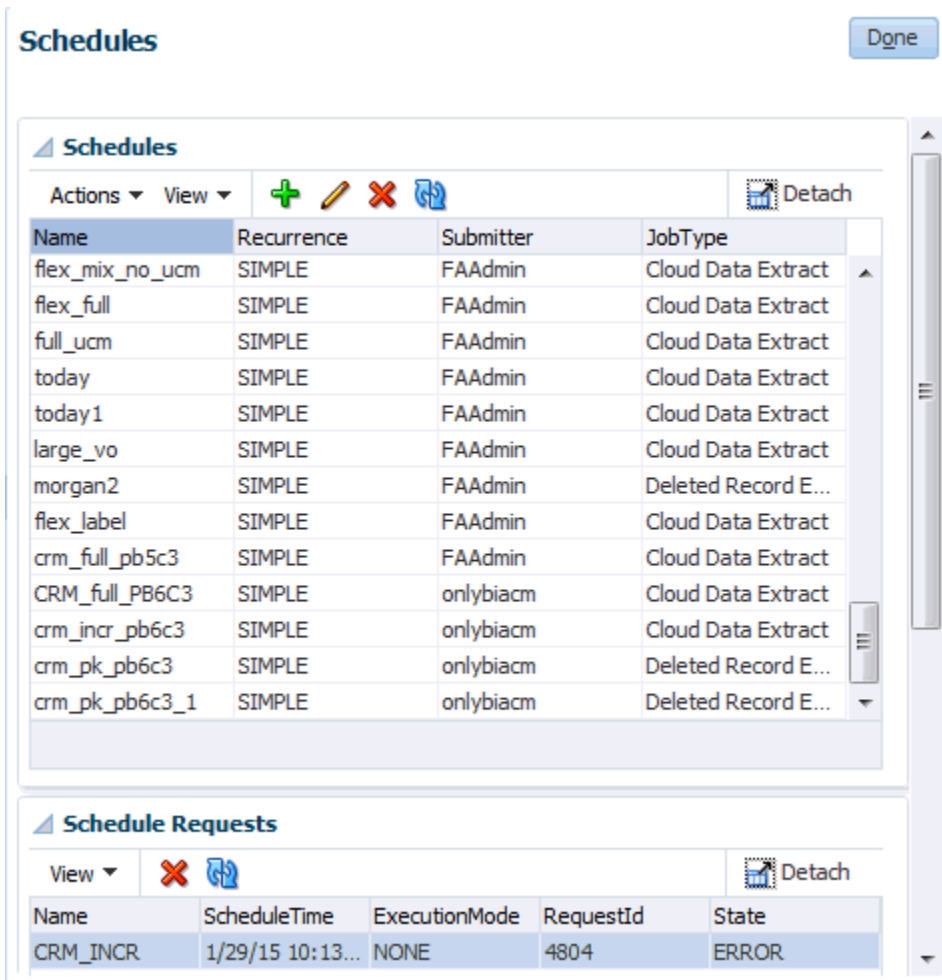
Synchronizing Deletes for a Cloud Extract

If you're loading data into the Universal Content Management (UCM) server using a Cloud Extract, then you use Oracle BI Cloud Connector Console (BICCC) to synchronize deletes in the source system with deletes in the Cloud Extract data.

Before you start, log into Oracle BI Cloud Connector Console using the Web link and user name supplied to you by Oracle Cloud Support. You can also start Oracle BI Cloud Connector Console using a HTTP URL based on the following format: `http://<FA OHS Host>:<FA OHS Port>/biacm`.






To synchronize your Cloud Extract data:

1. Select the **Configure Cloud Extract** link on the Tasks pane.
2. Click the Manage Extract Schedules icon () to display the Schedules dialog.






Schedules Done

Schedules


Actions ▾ View ▾      Detach

Name	Recurrence	Submitter	JobType
flex_mix_no_ucm	SIMPLE	FAAdmin	Cloud Data Extract
flex_full	SIMPLE	FAAdmin	Cloud Data Extract
full_ucm	SIMPLE	FAAdmin	Cloud Data Extract
today	SIMPLE	FAAdmin	Cloud Data Extract
today1	SIMPLE	FAAdmin	Cloud Data Extract
large_vo	SIMPLE	FAAdmin	Cloud Data Extract
morgan2	SIMPLE	FAAdmin	Deleted Record E...
flex_label	SIMPLE	FAAdmin	Cloud Data Extract
crm_full_pb5c3	SIMPLE	FAAdmin	Cloud Data Extract
CRM_full_PB6C3	SIMPLE	onlybiacm	Cloud Data Extract
crm_incr_pb6c3	SIMPLE	onlybiacm	Cloud Data Extract
crm_pk_pb6c3	SIMPLE	onlybiacm	Deleted Record E...
crm_pk_pb6c3_1	SIMPLE	onlybiacm	Deleted Record E...

Schedule Requests

View ▾    Detach

Name	ScheduleTime	ExecutionMode	RequestId	State
CRM_INCR	1/29/15 10:13...	NONE	4804	ERROR

3. Click the Add icon () to create a new schedule.
4. Use the **Name** field to specify a short name to identify this schedule in Oracle BI Cloud Connector Console.

5. Select 'Deleted Record Extract' from the **Job Type** drop down.
6. Use the **Recurrence** drop down to specify when you want to synchronize the data.

You typically schedule the synchronization to run at a time when the load on the Fusion Applications source system is low, for example, during off-peak hours. In addition, you must schedule the synchronization so that it doesn't conflict with the Cloud Extract or the data load into the Oracle Business Analytics Warehouse.

7. Save the details.

Configuring Proxy Settings for WebLogic Server

To use external web services or HTTP data sources when Oracle BI Applications is configured behind a firewall or requires a proxy to access the internet, you must configure Oracle WebLogic Server to allow the web service requests and to be aware of the proxy.

This task is part of the [Fusion Application Cloud Source System Deployment Roadmap](#).

Define the proxy host and the non-proxy hosts to WebLogic BI and ODI-managed server (bi_server1 and ODI_server1) by setting the following parameters:

- -Dhttp.proxyHost - specifies the proxy host. For example:
-Dhttp.proxyHost=www-proxy.example.com
- -Dhttp.proxyPort - specifies the proxy host port. For example:
-Dhttp.proxyPort=80
- -Dhttp.nonProxyHosts - specifies the hosts to connect to directly, not through the proxy. Specify the list of hosts, each separated by a "|" character; a wildcard character (*) can be used for matching. For example:
-Dhttp.nonProxyHosts="localhost|*.example1.com|*.example2.com"

To set these proxy parameters and the Web service configuration for your WebLogic Server:

1. Sign in to Weblogic Admin Console, click **Lock and Edit**, navigate to **Environments**, and expand **Servers**.
2. In Servers, click **bi_server1**, and select the Server_start tab.
3. In the Server_start tab, click **Arguments**, and enter the following in the Argument dialog:

 **Note:**

Don't erase existing arguments; add these lines along with the existing arguments.

```
-Dhttps.proxyHost=www.proxy.example.com -Dhttps.proxyPort=80
-Dhttp.proxyHost=www.proxy.example.com -Dhttp.proxyPort=80
-Dhttp.nonProxyHosts=localhost|*.mycompany.com|*.mycorporation.com|
*.otherhost
.com
-
Djavax.xml.soap.MessageFactory=oracle.j2ee.ws.saa.j.soap.MessageFacto
ryImpl
-
Djavax.xml.soap.SOAPFactory=oracle.j2ee.ws.saa.j.soap.SOAPFactoryImpl
-
Djavax.xml.soap.SOAPConnectionFactory=oracle.j2ee.ws.saa.j.client.p2p
.HttpSOAP
ConnectionFactory
```

where:

www-proxy.example.com is an example proxy host.

80 is the example proxy port.

localhost|*.mycompany.com|*.mycorporation.com|*.otherhost.com are example non-proxy hosts.

For hosts with http protocol, you can add -Dhttp.proxyHost and -Dhttp.proxyPort.

Managing Fusion Flex Extensions in Release 11.1.1.10.3

This section applies to the Oracle BI Applications release 11.1.1.10.3 and to Oracle Fusion Applications.

The warehouse tables, by default, carry a limited number of extension columns. For char type columns there are usually around 30 char columns in the warehouse tables that can be used for extensions. At times there could be cases where you end up with more than 30 columns. In such a case, only 30 columns would get mapped for extensions while the remaining columns get loaded in the SDS.

This section explains the current options available in such a case and provides a way by which you can control the columns that should get mapped rather than just let the algorithm decide which columns should get mapped.

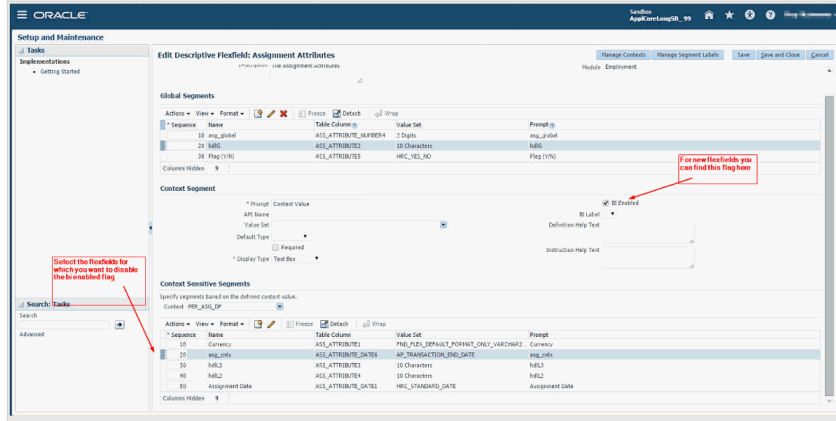
Option 1: Using the BI Applications Enabled Flag on the Fusion side

This is the recommended approach.


If all the flexfields you defined on the VO aren't relevant for BI Applications, you can restrict the flex extensions in BI Applications to only those that are relevant to BI Applications, by following these steps:

1. Enable the **BI Enabled** option.

When you define a flex field in Fusion, you see the **BI Enabled** option. Enabling this option ensures that this flex field gets extended to the repository and eventually flows into the BI Applications. However, if some columns aren't required for BI Applications, you can disable this option for those columns.



2. On editing the selected flex field, you can see the option for existing flexfields. Disable the option for the ones that aren't required for BI Applications.

 **Note:**

Disabling the **BI Enabled** option disables it for OTBI as well. Disable this option only if it isn't required for OTBI as well as BI Applications. There is currently no way to disable it only for BI Applications.

3. After disabling the option, wait until the BI Extender runs again to update the repository. Post that schedule, run a full extract, and do a full load again. Drop the SDS tables and reset the warehouse (using the option in the Configuration Manager) before running the full load.

Option 2: Specifying a Column Exclude List in ODI Repository:

If Option 1 isn't feasible, then try this option. The Extension framework in BI Applications looks at the ODI repository to identify which columns are extension columns. A workaround is to add the extension columns you don't need in BI Applications, to the ODI repository against the VO. This way the extension framework treats them as regular columns and not extension columns.

Based on your use case, follow either of these instructions:

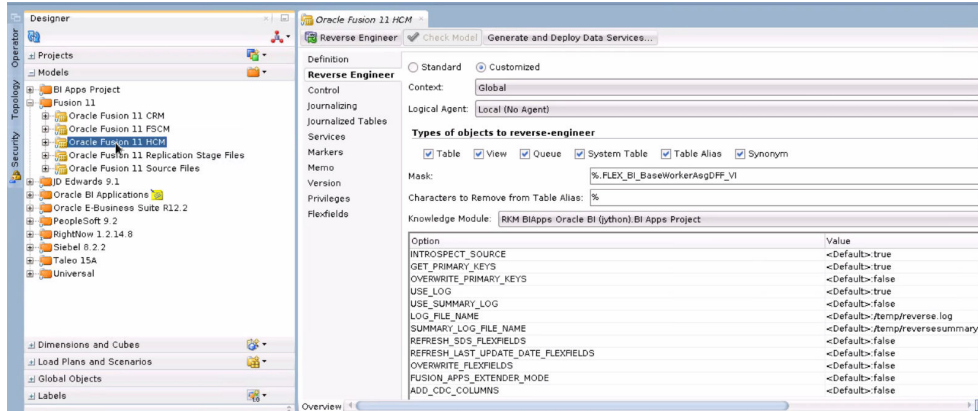
- You've access to the extended Fusion repository
- You don't have access to the extended Fusion repository

If you've access to the Fusion source repository from your ODI environment, follow these steps:

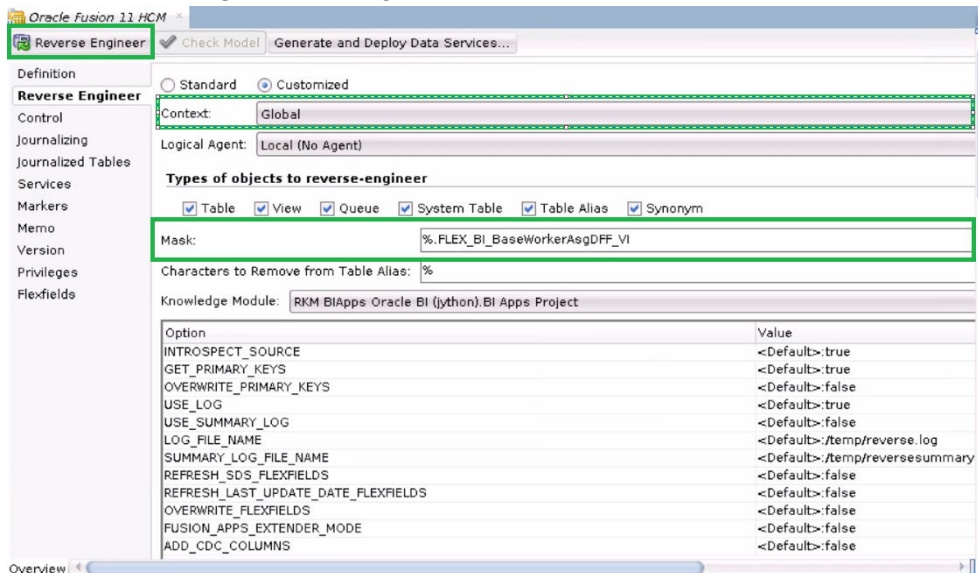
1. Run the Oracle BI Applications reverse knowledge module (RKM).
2. Log in to the ODI repository using the ODI studio, navigate to the **Models** tab, and open Oracle Fusion 11 HCM model as shown below.

 **Note:**

The Fusion Connection should be pointing to the BI Server with the extended repository.



3. Set the **Mask** property to `%.FLEX_BI_BaseWorkerAsgDFF_VI` as shown below. This prevents attempts to refresh the model for all the VOs.
4. Click **Reverse Engineer** to bring all the VO's extension columns into the model.



5. Once you see all the extension columns against the VO's definition, identify the extension columns that are needed for your business, and then delete the required columns from the data store.

For Example, if `SE_HRBP_` extension column is needed for your business, delete it and save the changes.

For the remaining extension columns which you don't need, or if you don't want the column to be populated or created in SDS, use the flexfield `OBI Populate Column` in SDS to remove them.

6. Edit the flexfield value to set the value as **N** (by default the flexfield value is be set to **Y**), and save the changes.

If you don't have access to the fusion source repository from your ODI Environment, follow these steps:

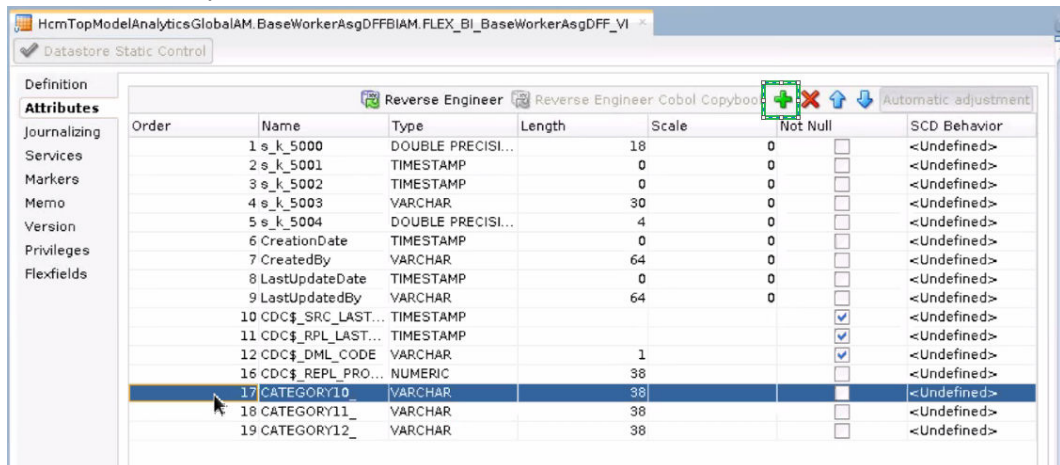
In this case, since you don't have access to the fusion source environment, there is no way to run RKM to reverse engineer a VO to add all extensible columns to it. Hence you should add all unused extensible columns manually against a VO's data store.



Note:

In Oracle Fusion Applications, the extractor provides mdcsv files which contain the metadata of the extended columns. Refer to that mdcsv file when you're manually adding the columns in the ODI.

1. Log in to the ODI repository using the ODI studio, navigate to the **Models** tab, find the data store, and open it as shown below.

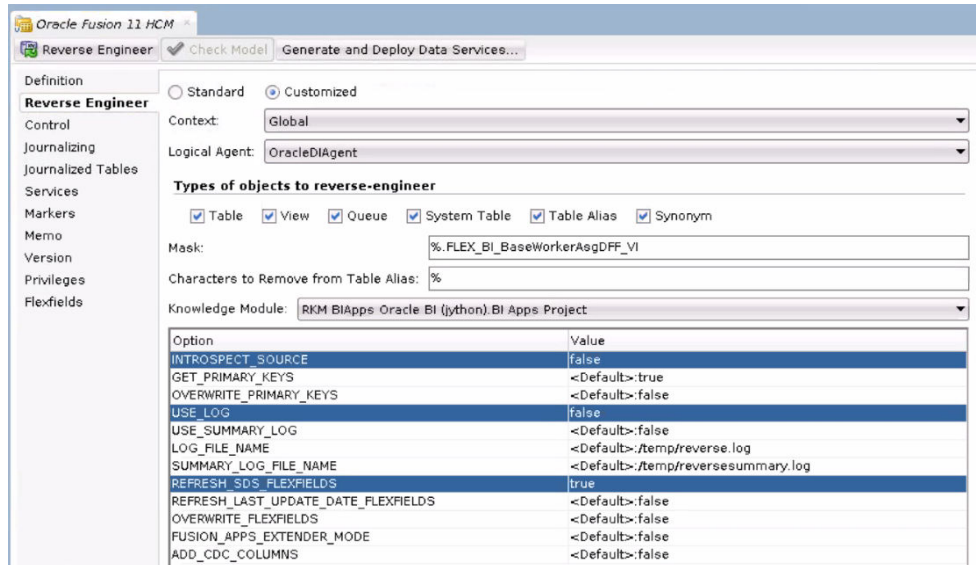


2. Click **Add** to add columns. Choose the correct data type, length and other options, and save the changes.

If you don't want the column to be populated or created in the SDS, then use the flexfield OBI Populate Column in SDS to remove them.

3. Edit the columns and go to the **Flexfields** tab. By default, the flexfield value is set to **Y**. Edit the flexfield value, set the value to **N**, and save the changes.
4. Run the RKM to update the newly added extensible column's short name.
5. Navigate to the **Models** tab and open the Oracle Fusion 11 HCM model.
6. Click the **Reverse Engineer** tab and set the **Mask** property to `% .FLEX_BI_BaseWorkerAsgDFF_VI` as shown below. This prevents attempts to refresh

the model for all the VOs.



7. Set the **INTROSPECT_SOURCE** option to **false**, the **USE_LOG** option to **false**, and enable the **REFRESH_SDS_FLEXFIELDS** mode by changing the option value to **true**.
8. Click **Reverse Engineer** to bring all VO extension columns into the model.

 **Note:**

If you've already loaded data, then you need to do a full load again. Reset the warehouse, make sure that all warehouse tables are empty, and reload.

Option 3

If Option 1 and 2 aren't possible, then use option 3. In this option you attempt to manually update the backend table that controls the mapping between the source column and the target warehouse column. Identify a column that has already been mapped but is not required, and then run backend update statements to switch it to a column that should get mapped instead.

As an example, assume that CATEGORY10_ extension column isn't required and you want to switch this mapping with INTERNATIONAL__STATUS_ extension column.

1. Connect to BIApps warehouse schema using SQL tools.
2. Run the following update script:

```
UPDATE W_ETL_FLEX_SQL SET FLEX_SRC_ATTRIB
='INTERNATIONAL__STATUS_',
ATTRIB_SQL_EXPRESSION='ASGDF.FLEX_BI_BaseWorkerAsgDFFBIAM.FLEX_BI_BaseWorkerAsgDFF_VI' AND FLEX_SRC_ATTRIB ='CATEGORY10_'
WHERE FLEX_CODE
='HRASG' AND FLEX_SRC_TABLE
='HcmTopModelAnalyticsGlobalAM.BaseWorkerAsgDFFBIAM.FLEX_BI_BaseWorkerAsgDFF_VI' AND FLEX_SRC_ATTRIB ='CATEGORY10_'
AND FLEX_ATTRIB='ASSIGNMENT_ATTR14_CHAR'
AND DATASOURCE_NUM_ID =200;
```



```
COMMIT;
```

 **Note:**

If the data is already loaded then you need to reload it to use the new mapping. Don't attempt to truncate all warehouse tables before doing the reload. Instead use the reset warehouse option in the Configuration Manager. Truncating all warehouse tables truncates the mapping table as well.

Setting Up Key Flex Fields for Financials Fusion Applications Source Data

If applicable, load the financial KFF data for GL#, CAT# and LOC# KFFs into Oracle Business Analytics Warehouse.

These instructions only apply if you're deploying a Fusion Applications cloud data source.

1. Enable and set up the VOs for extraction in Oracle BI Cloud Connector Console. See [Using Oracle BI Cloud Connector Console to Configure Cloud Extract](#).
2. Configure the ODI repository in ODI. See [Using ODI to Set Up Key Flex Fields](#).

Using Oracle BI Cloud Connector Console to Configure Cloud Extract

When you've completed the key flexfield (KFF) setup in Oracle Fusion Applications and deployed the flexfield, view objects (VO) are generated for each segment that you've enabled for Oracle Business Intelligence (BI). Each VO must be seeded and enabled for BI extraction so that it can be loaded into Oracle Business Analytics Warehouse.

To seed and enable VOs for extraction:

1. In Oracle BI Cloud Connector Console, navigate to the **Configure Cloud Extract** link on the Tasks pane to display the Configure Cloud Extract dialog.
2. Scroll down to the Data Store area.
3. Click **Add** to display the Add Data Store dialog, specify the details, then click **Save** and **Close**.

 **Note:**

VO names are case-sensitive.

You must add VOs in the **GL Journals** under the **Oracle Financial Analytics Offering**, as shown in the example screen shot.

4. To enable VOs for extraction, select the parent Offering, Functional Area, FG/DG, or individual VO.
5. Enable and add these VOs for extraction.
 - For the GL# KFF, you must add the BI Flattened VO along with the segment VOs generated for the segment labels mapped to BI Objects in your Fusion setup.

The BI Flattened VO name for GL# that must be added is
FscmTopModelAM.AccountBIAM.FLEX_BI_Account_VI.

- The VOs generated for segment labels FA_COST_CTR (mapped to Dim – Cost Center), GL_ACCOUNT (mapped to Dim – Natural Account Segment), and GL_BALANCING (mapped to Dim – Balancing Segment) are known prior. You must add these VOs:

```
FscmTopModelAM.AccountBIAM.FLEX_TREE_VS_FA_COST_CTR_VI
FscmTopModelAM.AccountBIAM.FLEX_TREECODE_VS_FA_COST_CTR_VI
FscmTopModelAM.AccountBIAM.FLEX_TREE_VS_GL_BALANCING_VI
FscmTopModelAM.AccountBIAM.FLEX_TREECODE_VS_GL_BALANCING_VI
FscmTopModelAM.AccountBIAM.FLEX_TREE_VS_GL_ACCOUNT_VI
FscmTopModelAM.AccountBIAM.FLEX_TREECODE_VS_GL_ACCOUNT_VI
```

 **Note:**

For these VOs, ensure that you enable all the columns for the select list.

When you add the TREECODE view objects, you must enable the **Disable Effective date filter** option. Additionally, ensure that `invunitsofmeasurep` view object is disabled for Human Capital Management. For Fusion Supply Chain Management and Customer Relationship Management, deploy the `INV_UOM_CLASSES` descriptive flexfield for `invunitsofmeasurep` view object.

- The VOs generated for any segment label apart from FA_COST_CTR, GL_BALANCING and GL_ACCOUNT, and mapped to the BI Object Dim – GL Segmentx aren't known before hand as the VO names depends on the segment label. Therefore, you must add the VO names based on the naming pattern.

If the segment is a tree segment (if you've created hierarchies for value sets used for those segments), then the VOs are named:

```
FscmTopModelAM.AccountBIAM.FLEX_TREE_VS_<segment label>_VI
```

FscmTopModelAM.AccountBIAM.FLEX_TREECODE_VS_<segment label>_VI

If the segment is a non-tree segment (if you've NOT created hierarchies for value sets used for those segments), then the VOs are named:

FscmTopModelAM.AccountBIAM.FLEX_VS_<segment label>_VI

For example:

Segment Label	Tree v/s Non Tree	VO Name
PRODUCT	Tree	FscmTopModelAM.AccountBIAM.FLEX_TRE E_VS_PRODUCT_VI FscmTopModelAM.AccountBIAM.FLEX_TRE ECODE_VS_PRODUCT_VI
ORG	Non Tree	FscmTopModelAM.AccountBIAM.FLEX_VS_ ORG_VI

 **Note:**

When you add TREECODE VOs, you must enable the **Disable Effective date filter** option.

For the CAT# and LOC# KFFs, you need to add the BI Flattened VO for each KFF.

The BI Flattened VO name for CAT# that needs to be added is

FscmTopModelAM.CategoryBIAM.FLEX_BI_Category_VI. The BI Flattened VO name for LOC# that needs to be added is

FscmTopModelAM.LocationBIAM.FLEX_BI_Location_VI.

- For Units of Measure (UOM) extensions, add the FscmTopModelAM.InvUomPublicViewAM.InvUnitsOfMeasurePVO view object.

Once you've added these VOs and completed the extraction process, the data files for these VOs are downloaded into the physical location configured for replication stage files logical schema in ODI (Fusion 9: DS_FUSION_9_0_REPLSTG_FILES, Fusion 10: DS_FUSION_10_0_REPLSTG_FILES).

The files generated for GL# segment labels mapped to Dim – GL Segmentx (Dim – GL Segment 1-10) have corresponding pre-seeded data stores using a generic naming convention in ODI. The next step is to rename the corresponding files with appropriate file names so that they can load the corresponding data stores.

The mapping between the files generated and the new file names corresponding to the data stores is based on the segment label to BI Object mapping and setup for that segment (tree versus non-tree).

Taking the same example as mentioned in the previous steps, – PRODUCT label is mapped to Dim – GL Segment 1 and is a tree segment, and ORG label is mapped to Dim – GL Segment2 and is a non tree segment, then the mapping between the generated files and the new files are:

File Generated	New File Name
File_FscmTopModelAM.AccountBIAM.FLEX_TR EE_VS_PRODUCT_VI	File_FscmTopModelAM.AccountBIAM.FLEX_TR EE_VS_GL_SEGMENT1_VI

File Generated	New File Name
File_FscmTopModelAM.AccountBIAM.FLEX_TR EECODE_VS_PRODUCT_VI	File_FscmTopModelAM.AccountBIAM.FLEX_TR EECODE_VS_GL_SEGMENT1_VI
File_FscmTopModelAM.AccountBIAM.FLEX_VS _ORG_VI	File_FscmTopModelAM.AccountBIAM.FLEX_VS _GL_SEGMENT2_VI

This mapping has to be provided in a configuration file named `FinKFFFileRename.cfg`, which is used in the downstream processes to load the data from those files.

6. Create and save this configuration file in the same location with all the data files generated in this procedure.

This configuration file has to be placed in the replication staging location in the ODI environment (the location where the files are copied and placed from UCM to ODI).

The configuration file has to be created as a comma delimited file, and the format of the entries in the file has to be as shown in this sample entry. Save the file when you're done making the entries.

<Generated File Name>,<New File Name based on the pre seeded data store>

A sample entry in the file would look like this:

```
file_fscmtopmodelam_accountbiam_flex_vs_gl_management_vi,file_fscmtopmodelam_accountbiam_flex_vs_gl_segment2_vi
```

Create the configuration file in the same platform (Windows or Linux) where you'll eventually save this file. Avoid copy and paste from one platform to the other. This avoids issues regarding special characters in text files when crossing different platforms.

Using ODI to Set Up Key Flex Fields

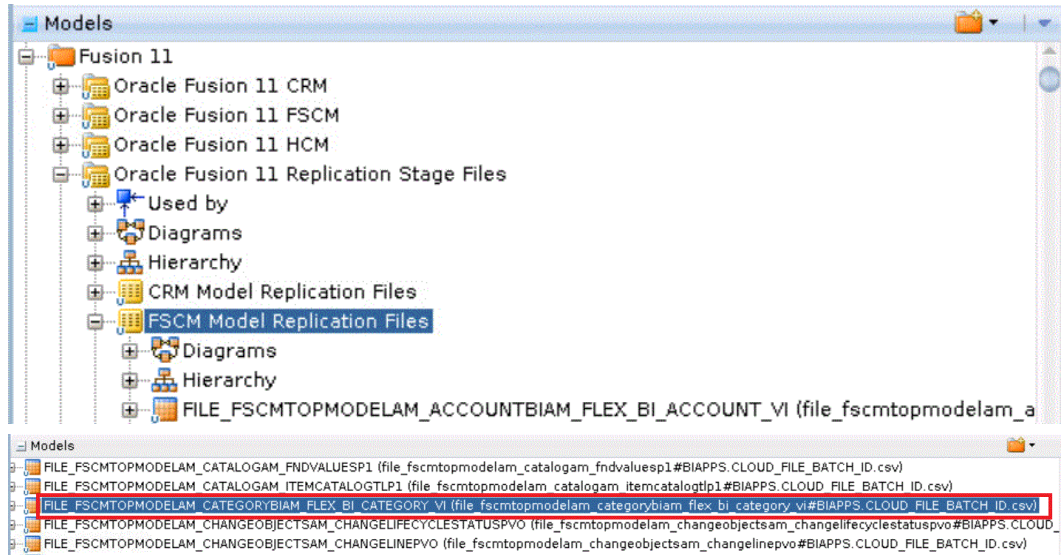
The BI (business intelligence) flattened VO (view object) generated for each key flexfield (KFF) has columns relating to each segment label. Use this procedure to set up the mappings for these columns.

1. For CAT# KFF, check the csv file generated for the BI flattened VO for the list of additional columns for each segment -
`file_fscmtopmodelam_categorybiam_flex_bi_category_vi%.csv`.

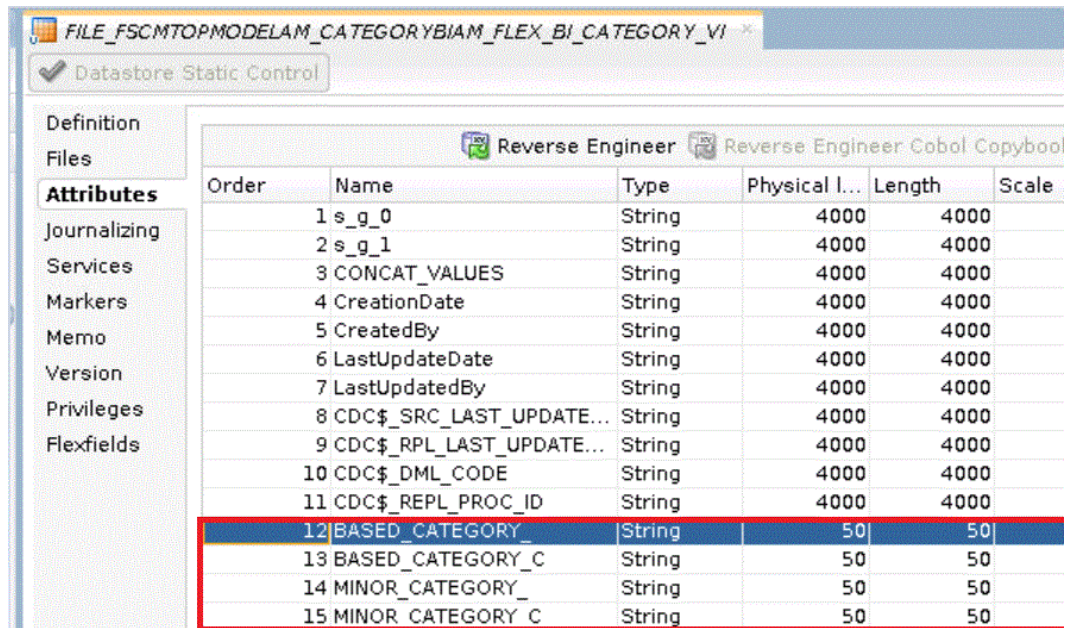
These segment columns end with “_” or “_c”. For example, `BASED_CATEGORY_`, `MINOR_CATEGORY_`.

You must add these new columns in the file data store and the VO data store in ODI.

Navigate to the file data store to add the new columns.



2. Add the new columns to the FILE_FSCMTOPMODELAM_CATEGORYBIAM_FLEX_BI_CATEGORY_VI data store with the default data type of String (50,50) for all columns.

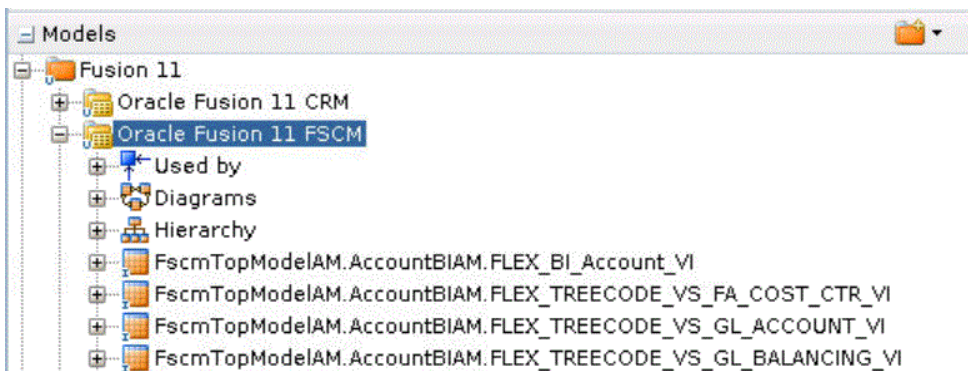


For LOC# KFF, check the CSV file generated for the BI flattened VO for the list of additional columns for each segment - file_fscmtopmodelam_locationbiam_flex_bi_location_vi%.csv. These segment columns end with “_” or “_c”. The corresponding file data stores for LOC# KFF is FILE_FSCMTOPMODELAM_LOCATIONBIAM_FLEX_BI_LOCATION_VI.

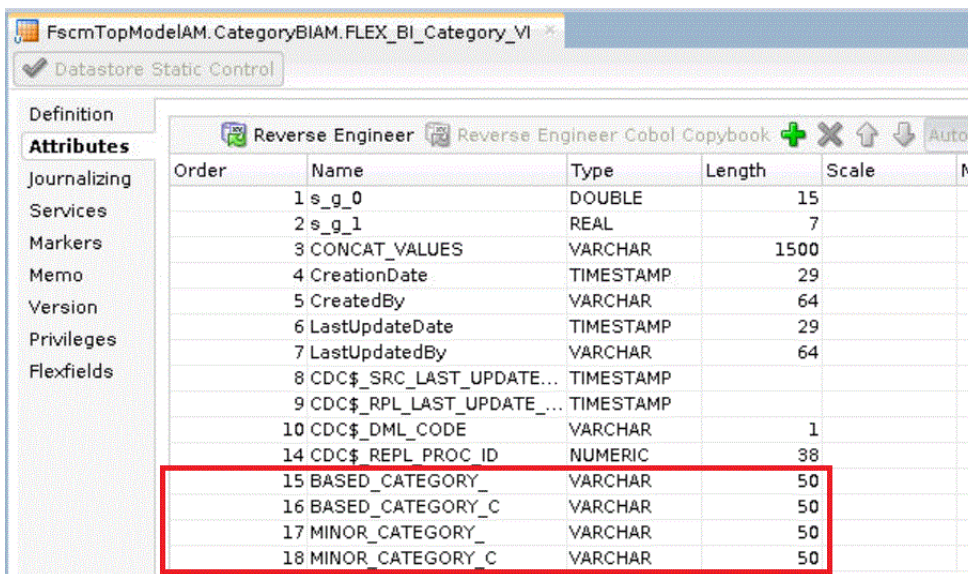
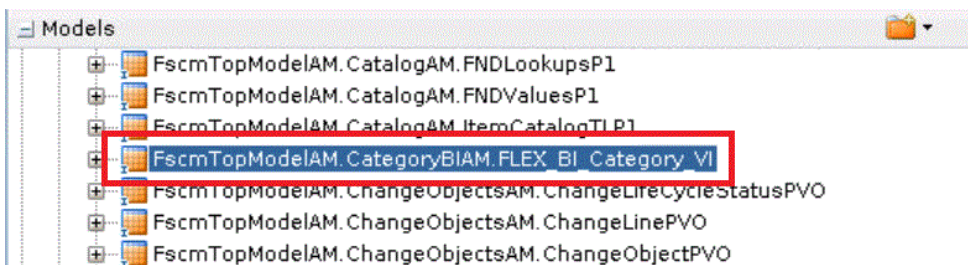
3. Repeat the configuration steps for the corresponding VO data stores in ODI.

The VO data stores are in the path shown in the image. Use the default data type of Varchar(50, 50).

- CAT# KFF VO data store: FscmTopModelAM.CategoryBIAM.FLEX_BI_Category_VI
- LOC# KFF VO data store: FscmTopModelAM.LocationBIAM.FLEX_BI_Location_VI



Scroll down to locate the VO data stores.

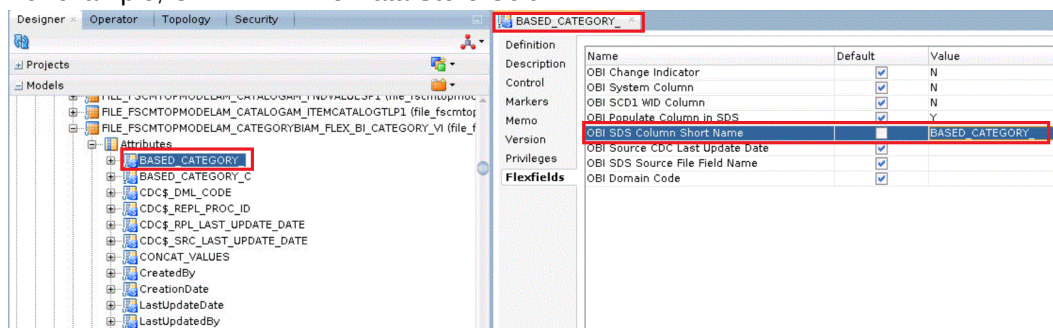


- For the newly added columns in both the file data store and VO data store, set the flexfield value for **OBI SDS Column Short Name**.

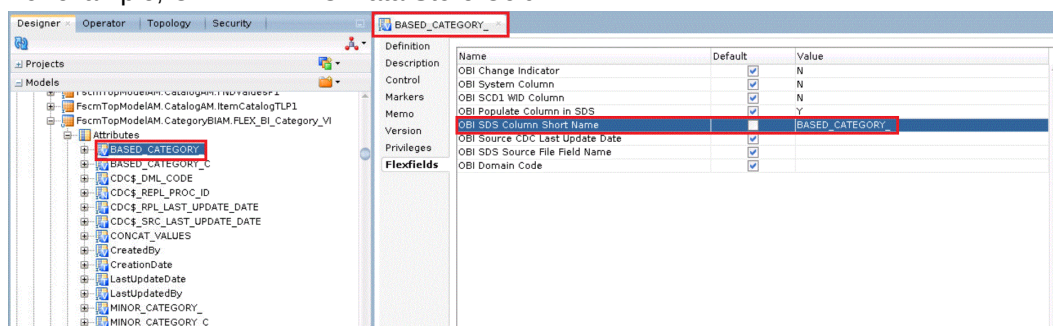
Ensure that the flexfield value is the same as the column name. Also, ensure that you set the flexfield value **ODI Populate Column** to **Y** in SDS for these columns.

These images display example flexfield values.

For example, CAT# KFF File Data Store Column:



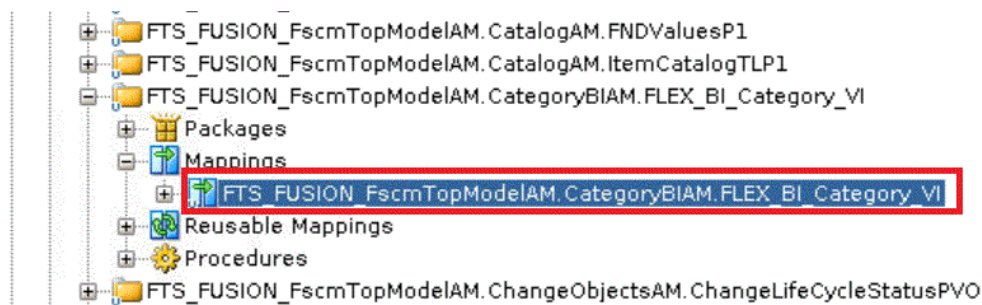
For example, CAT# KFF VO Data Store Column:



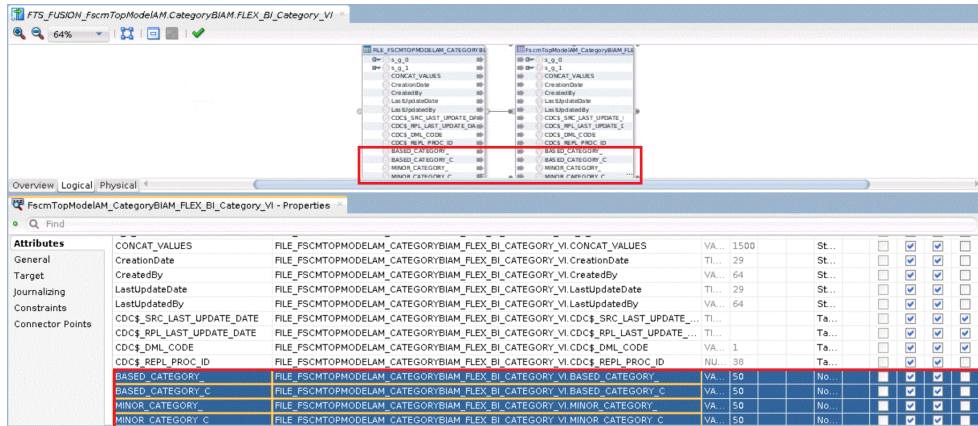
- Repeat for LOC# KFF.
- Execute `GENERATE_SDS_DDL` in `INCR_REFRESH_MODE`.

This process updates the SDS table to include the new extended columns.

- When you have added the columns in the appropriate data stores, map the columns in the corresponding ODI mappings to load the data.
- For the CAT# KFF, edit the FTS mapping located under the `SDE_FUSION_V1_ADAPTOR` folder to map these new columns.



- Create a one-to-one mapping between the segment columns in `FTS_FUSION_FscmTopModelAM.AccountBIAM.FLEX_BI_Account_VI`.



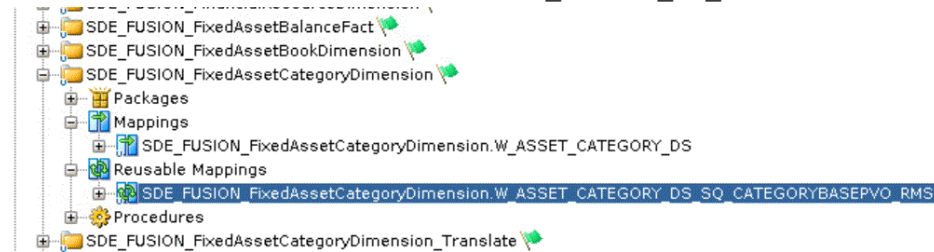
10. Save the changes, then regenerate the underlying scenario for the package.
11. Similarly, if you have LOC# KFF, edit the corresponding FTS mapping to map the new columns:

LOC# KFF FTS mapping:

FTS_FUSION_FscmTopModelAM.LocationBIAM.FLEX_BI_Location_VI.

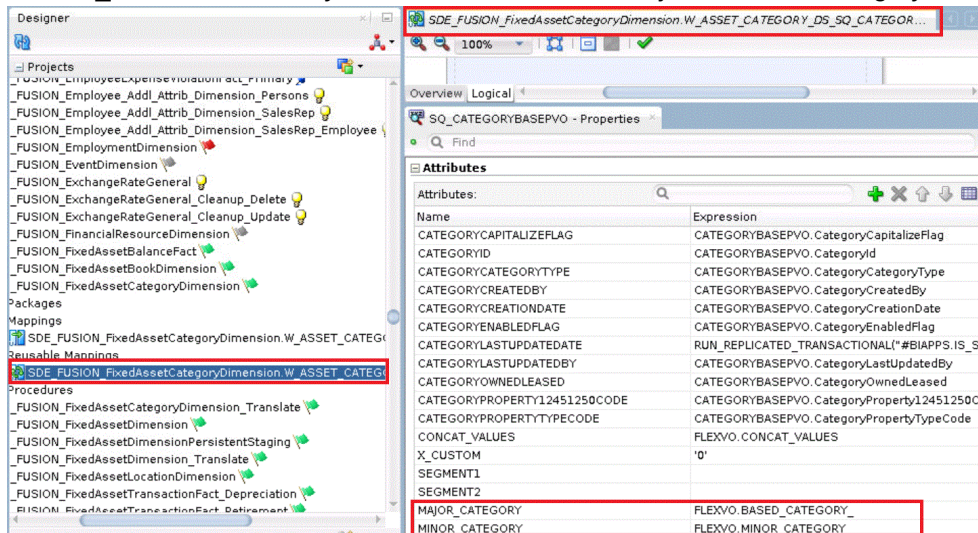
12. Save the changes and regenerate the underlying scenarios for the packages.
13. Edit the SDE mapping for CAT# KFF to map these new columns to corresponding columns in the staging table.

The SDE mapping is located in the same SDE_FUSION_V1_Adaptor folder.



The column mapping is based on the segment label to BI Object mapping done in your Fusion Applications configuration.

For CAT# KFF, if you have mapped BASED_CATEGORY and MINOR_CATEGORY then you can use them as the Major and Minor category.



14. Save the changes and regenerate the underlying scenario for the package.
15. If applicable, complete similar steps for LOC# KFF based on the columns that you extend for these:
 - SDE mapping: SDE_FUSION_FixedAssetLocationDimension
 - Staging table: W_ASSET_LOCATION_DS

Setting Up Key Flex Fields for HCM Fusion Applications Source Data

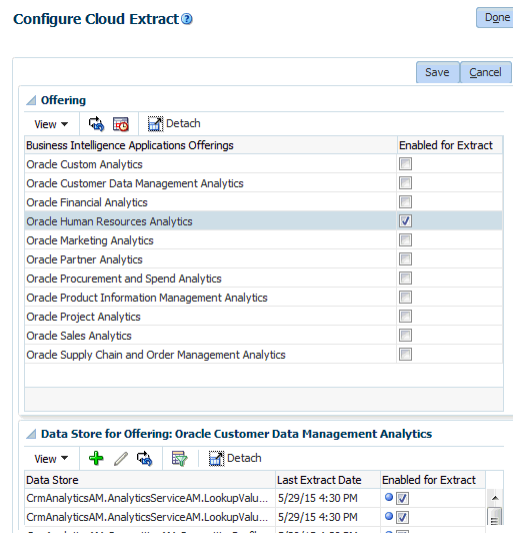
If applicable, load the HCM KFF data mentioned in this topic into Oracle Business Analytics Warehouse.

These instructions only apply if you're deploying a Fusion Applications cloud data.

When you've completed the KFF setup in Fusion and deployed the Flexfield, VOs are generated for each segment that you've enabled for BI. Each VO must be seeded and enabled for BI extraction so that it can be loaded into Oracle Business Analytics Warehouse.

To set up key flex fields for HCM Fusion applications source data:

- Use Oracle BI Cloud Connector Console to load the View Objects (VOs) listed in Supported View Objects below:
 1. On the Fusion Applications pod, start Oracle BI Cloud Connector Console using a HTTP URL based on the following format: `http://<FA OHS Host>:<FA OHS Port>/biacm`
 2. In Oracle BI Cloud Connector Console, select the **Configure Cloud Extract** link on the Tasks pane to display the Configure Cloud Extract dialog.



3. Scroll down to the Data Store area.
4. For each of the View Objects (VOs) listed in the Supported View Objects table below, click the **Add** icon (+) to display the Add Data Store dialog and create a new data store.

The screenshot shows a dialog box titled "Add Dialog" with a close button (X) in the top right corner. Inside the dialog, there are two main sections:

- Add Data Store:** This section contains three fields:
 - * Data Store Key: A text input field.
 - * Enabled: A checkbox.
 - * Disable Effective date filter: A checkbox.
- Associate Data Store with Fact Group or Dimension Group:** This section contains two dropdown menus:
 - * Group Type: A dropdown menu with "Fact Group" selected.
 - * Group Name: A dropdown menu with "AP Holds" selected.

At the bottom of the dialog, there are two buttons: "Save and Close" and "Cancel".

 **Note:**

FscmTopModelAM.InvUomPublicViewAM.InvUnitsOfMeasurePVO view object could fail if you haven't deployed the INV_UOM_CLASSES flexfield. If you encounter a failure, then disable this view object for Human Capital Management using Oracle BI Cloud Connector Console.

The next scheduled Cloud Extract executed by Oracle BI Cloud Connector Console extracts these VOs and generates files as required by you. Then, the next scheduled load plan executed by Configuration Manager downloads the newly generated FLEX files and loads the flexfield changes into the Oracle Business Analytics Warehouse.

Supported View Objects

Presentation Table Name in Fusion Applications	Target Table	Model Name	VO Name
Assignment Extensible Attributes	W_HR_ASSIGNMENT_DS/D	Dim – HR Assignment	HcmTopModelAnalyticsGlobalAM.BaseWorkerAsgDFFBIAM.FLEX_BI_BaseWorkerAsgDFF_VI

Presentation Table Name in Fusion Applications	Target Table	Model Name	VO Name
Assignment History Extensible Attributes	W_HR_ASSIGN_ADDL_D	Dim – HR Assignment History	HcmTopModelAnalyticsGlobalAM.BaseWorkerAsgDFFBIAM.FLEX_BI_BaseWorkerAsgDFF_VI HcmTopModelAnalyticsGlobalAM.BaseWorkerAsgLegDDFBIAM.FLEX_BI_BaseWorkerAsgLegD DF_VI HcmTopModelAnalyticsGlobalAM.WorkRelationshipDFFBIAM.FLEX_BI_WorkRelationshipDFF_VI HcmTopModelAnalyticsGlobalAM.WorkRelationshipLegDDFBIAM.FLEX_BI_WorkRelationshipLegDDF_VI HcmTopModelAnalyticsGlobalAM.ContractDFFBIAM.FLEX_BI_ContractDFF_VI HcmTopModelAnalyticsGlobalAM.ContractLegDDFBIAM.FLEX_BI_ContractLegDDF_VI
Person Extensible Attributes	W_HR_PERSON_DS/D	Dim – HR Person	HcmTopModelAnalyticsGlobalAM.PersonsDFFBIAM.FLEX_BI_PersonsDFF_VI HcmTopModelAnalyticsGlobalAM.PersonDetailsDFFBIAM.FLEX_BI_PersonDetailsDFF_VI HcmTopModelAnalyticsGlobalAM.PersonTypeUsagesDFFBIAM.FLEX_BI_PersonTypeUsagesD FF_VI

Presentation Table Name in Fusion Applications	Target Table	Model Name	VO Name
Person Extensible Attributes	W_HR_PERSON_LEG_DS/D	Dim – HR Person Legislation	HcmTopModelAnalyticsGlobalAM.PersonLegislativeInfoDFBIAM.FLEX_BI_PersonLegislativeInfoDFI_VI HcmTopModelAnalyticsGlobalAM.CitizenshipDFBIAM.FLEX_BI_CitizenshipDFI_VI HcmTopModelAnalyticsGlobalAM.EthnicityDFBIAM.FLEX_BI_EthnicityDFI_VI HcmTopModelAnalyticsGlobalAM.PersonLegDFBIAM.FLEX_BI_PersonLegDFI_VI HcmTopModelAnalyticsGlobalAM.PersonNameDFBIAM.FLEX_BI_PersonNameDFI_VI HcmTopModelAnalyticsGlobalAM.PersonNameLegDFBIAM.FLEX_BI_PersonNameLegDFI_VI HcmTopModelAnalyticsGlobalAM.ReligionDFBIAM.FLEX_BI_ReligionDFI_VI HcmTopModelAnalyticsGlobalAM.PersonVisaLegDFBIAM.FLEX_BI_PersonVisaLegDFI_VI HcmTopModelAnalyticsGlobalAM.VisaPermitDFBIAM.FLEX

Presentation Table Name in Fusion Applications	Target Table	Model Name	VO Name
			_BI_VisaPermitDFF_VI
Position Extensible Attributes	W_HR_POSITION_DS/D	Dim – HR Position	HcmTopModelAnalyticsGlobalAM.PositionCustomerFlexBIAM.FLEX_BI_PositionCustomerFlex_VI
Pay Grade Extensible Attributes	W_PAY_GRADE_DS/D	Dim – Pay Grade	HcmTopModelAnalyticsGlobalAM.GradeCustomerFlexBIAM.FLEX_BI_GradeCustomerFlex_VI
Job Extensible Attributes	W_JOB_DS/D	Dim – Job	HcmTopModelAnalyticsGlobalAM.JobCustomerFlexBIAM.FLEX_BI_JobCustomerFlex_VI
Location Extensible Attributes	W_BUSN_LOCATION_DS/D	Dim - Worker Location	HcmTopModelAnalyticsGlobalAM.LocationCustomerFlexBIAM.FLEX_BI_LocationCustomerFlex_VI
Department Extensible Attributes	W_INT_ORG_DS/D	Dim - Department	
Payroll Extensible Attribute	W_PAY_PAYROLL_D	Dim - Payroll Details	HcmTopModelAnalyticsGlobalAM.AllPayrollsDFFBIAM.FLEX_BI_AllPayrollsDFF_VI
Candidate Extensible Attributes	W_SCCN_CANDIDATE_D	Dim - Succession Candidate	HcmTopModelAnalyticsGlobalAM.PlanCandidateDFFBIAM.FLEX_BI_PlanCandidateDFF_VI

Presentation Table Name in Fusion Applications	Target Table	Model Name	VO Name
Workforce Event Type Extensible Attributes	W_WRKFC_EVENT_TYP E_D W_WRKFC_EVNT_ADDL_D W_WRKFC_EVNTRSN_ADDL_D	Dim - HR Workplace Event Type	HcmTopModelAnalyticsGlobalAM.ActionReasonsDFFBIAM.FLEX_BI_ActionReasonsDFF_VI HcmTopModelAnalyticsGlobalAM.ActionsDFFBIAM.FLEX_BI_ActionsDFF_VI HcmTopModelAnalyticsGlobalAM.ActionsLegDDFBIAM.FLEX_BI_ActionsLegDDF_VI
Project Attribute Extension	PROJECT_ADDL_D	Dim - Project	FscmTopModelAM.PJF_PROJECTS_ALLBIAM.FLEX_BI_PJF_PROJECTS_ALL_VI
Task Attribute Extension	TASK_ADDL_D	Dim - Task	FscmTopModelAM.TaskDffBIAM.FLEX_BI_TaskDff_VI
N/A	Employee_ADDL_D	Dim - Employee	
N/A	W_UOM_CONVERSION_G	N/A	

6

Setting Up Oracle Taleo Cloud Data Source

Follow this procedure to deploy Oracle Business Intelligence Applications (Oracle BI Applications) with Oracle Taleo Cloud source system.

Topic

- [Registering a Taleo Cloud Source System](#)

Registering a Taleo Cloud Source System

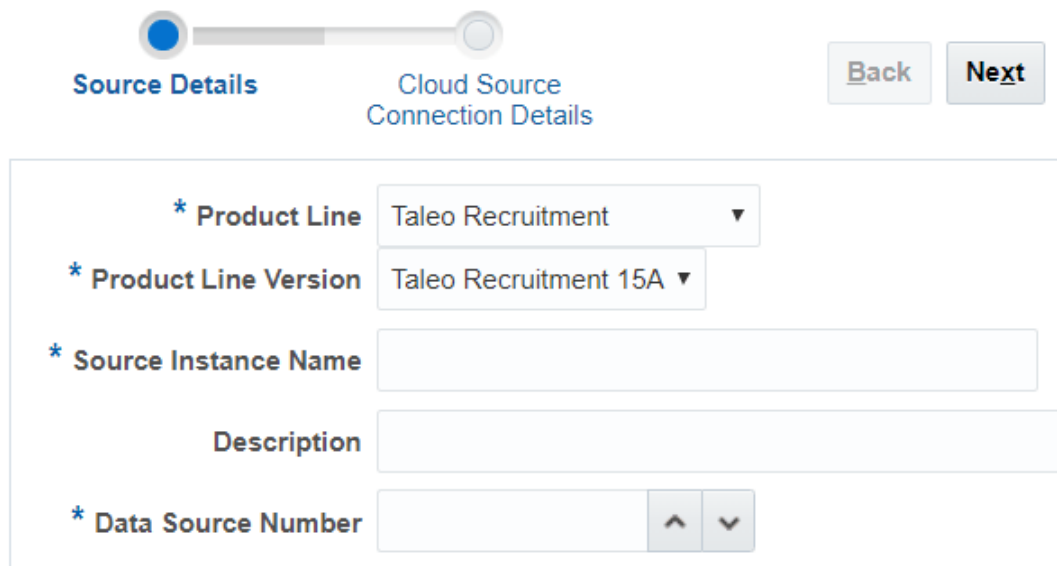
Use these instructions to register the applicable Taleo Cloud source system.

To register Taleo Cloud source system:

You must have the Taleo Cloud source details.

1. Log in to Oracle BI Applications Configuration Manager as a user that has Enterprise Role **-ASM_APPLICATION_IMPLEMENTATION_ADMIN_ABSTRACT**.
2. In Configuration Manager, select the **Define Business Intelligence Applications** link in the System Setups area on the Tasks pane.
3. Click the **Add** icon (+) to register a new source system using the Register Source dialog.
4. At the Register Source dialog, specify the following details, then click **Next**:

Register Source



The screenshot shows the 'Register Source' dialog box. At the top, there is a progress bar with two steps: 'Source Details' (which is active, indicated by a blue circle) and 'Cloud Source Connection Details'. To the right of the progress bar are two buttons: 'Back' and 'Next'. Below the progress bar, the form contains the following fields:

- * Product Line**: A dropdown menu with 'Taleo Recruitment' selected.
- * Product Line Version**: A dropdown menu with 'Taleo Recruitment 15A' selected.
- * Source Instance Name**: A text input field.
- Description**: A text input field.
- * Data Source Number**: A text input field with up and down arrow buttons.

- Product Line — Select your source system type. For example, Taleo Recruitment.
- Product Line Version — Select the data source version that is deployed.
- Source Instance Name — Enter a short readable name to identify this data source in Configuration Manager.

- Description — (Optional) Enter a short description to help Administrators and Implementers identify and use this data source in Configuration Manager.
 - Data Source Number — Enter an integer or use the spinner to specify a number to identify data in the OTBI-Enterprise data warehouse. This number must be unique within Configuration Manager. The number 999 is reserved, and shouldn't be used.
5. At the Cloud Replicator Details dialog, specify connection properties, then click **Save**:

Register Source

Source Details Cloud Source Connection Details Back Next

Replicator Schema Configuration

* SDS Data Source: jdbc/sds/CloudrepTaleo15ASdsDBDS Test SDS Connection

Cloud Source Connection Properties Test Cloud Source Connection

Name	* Value
DAILY_EXPORT_LIMIT	0
EXPORT_MAX_RECORDS	100000
HOSTNAME	
USER	
PASSWORD	

Save Cancel

 **Note:**

Taleo enforces daily and transaction limits for bulk data export requests. During a Taleo source connection registration, the default values are displayed (**DAILY_EXPORT_LIMIT=500000**, **EXPORT_MAX_RECORDS=100000**. If the Taleo zone limits are increased, it's recommended to update the Daily Limit on the Replicator to match the zone value or update it to 0.

DAILY_EXPORT_LIMIT=0 (0 indicates unlimited).

During an LP execution, if a replication job hits the Taleo Daily Limit, then the job is paused for 24 hours but the status of the ODI LP step is marked as Completed. Subsequent LP steps of replication for Taleo source would complete without actually running the replication and rest of the ETL steps (SDE, SIL) are also completed for Taleo source without running the ETL. The next day, LP triggers the rest of the replication and completes the ETL. The paused job is triggered automatically and runs the replication only for the entity which first hits the daily limit. For replicating rest of the entities, a new LP instance needs to be triggered after 24 hours of the previous LP.

7

Deploying the ODI Repository for Non-Oracle Source Databases

This section provides information about deploying the Oracle Data Integrator (ODI) repository for non-Oracle source databases.

During installation and setup of Oracle Business Intelligence Applications (Oracle BI Applications), if your source database is Oracle, then an instance of the ODI repository is automatically deployed and no further setup steps are required. If your source database is Microsoft SQL Server or UDB DB2, then you must perform these configuration steps to deploy a new instance of the ODI repository.

If you've multiple sources and one or more of the source databases is non-Oracle, you must follow these procedures for all source databases, including any Oracle source databases. The following non-Oracle source databases are certified:

- JD Edwards (JDE) 92 is certified with DB2 UDB (DB2 v10.5) and Microsoft SQL Server 2016 source databases.
- People Soft 92 HCM is certified with Microsoft SQL Server 2016 source databases.
- People Soft 92 LM (Learning Management) is certified with Microsoft SQL Server 2016 source databases.
- People Soft 92 FSCM is certified with Microsoft SQL Server 2016 source databases.
- People Soft 92 SIA (Student Information Analytics) is certified with Microsoft SQL Server 2016 source databases.

Perform these steps in order:

1. [Export Security Settings from Original ODI Repository.](#)
2. [Exporting Topology Settings from Original ODI Repository.](#)
3. [Create a New Master ODI Repository.](#)
4. [Connect to the New Master ODI Repository.](#)
5. [Create a New Work Repository.](#)
6. [Import Source Adaptor Metadata Content into the New ODI Repository.](#)
7. [Import Security Settings Into the ODI Repository.](#)
8. [Importing Connection Details Which Configures the Data Servers.](#)

Next Steps:

In Oracle BI Applications Configuration Manager, register sources, configure parameters, and generate load plans.

 **Note:**

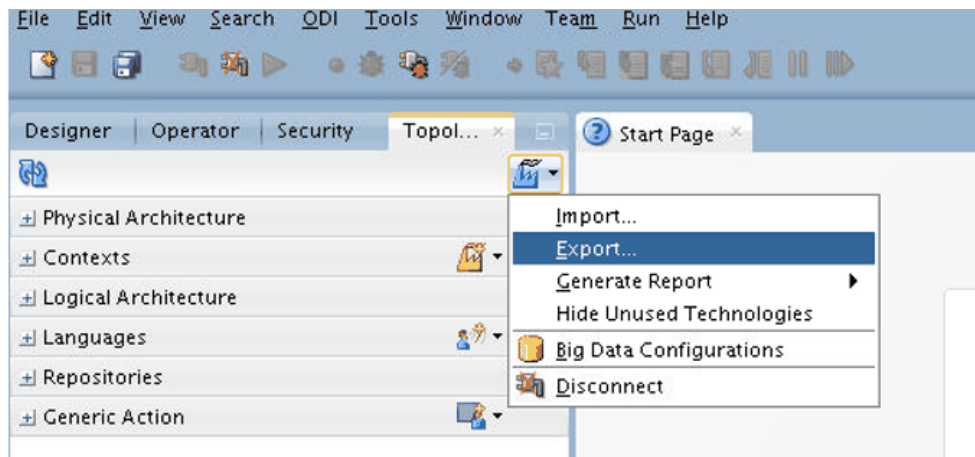
After you perform these procedures for a particular source and you then want to add a new source, you only need to perform the steps in [Importing Source Adapter Metadata Content into the New ODI Repository](#). This is true regardless of the database technology of the second source.

For example, if you've a PeopleSoft 9.2 source on a Microsoft SQL Server database, then you'd perform the steps in [Exporting Security Settings from Original ODI Repository](#) through [Importing Connection Details Which Configures the Data Servers](#). If you then wanted to add a second source, for example JD Edwards (on an Oracle, DB2, or Microsoft SQL Server database), you'd only need to perform the steps in [Importing Source Adapter Metadata Content into the New ODI Repository](#).

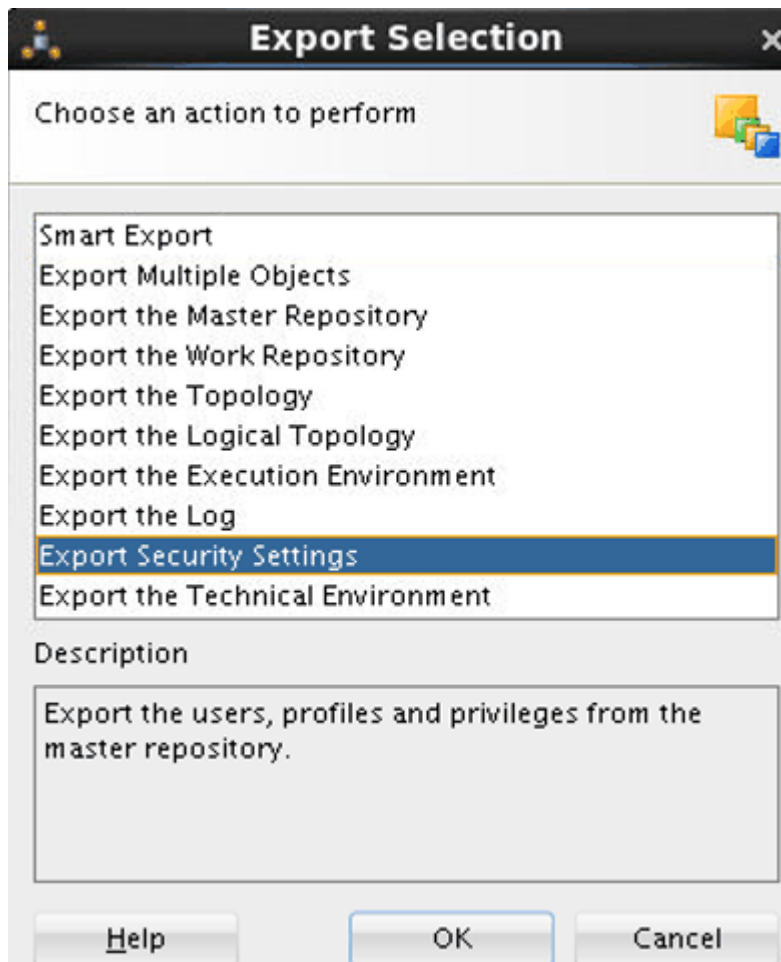
Exporting Security Settings from Original ODI Repository

You need to export the security settings from the ODI Repository that were created using the Oracle Business Analytics Suite RCU.

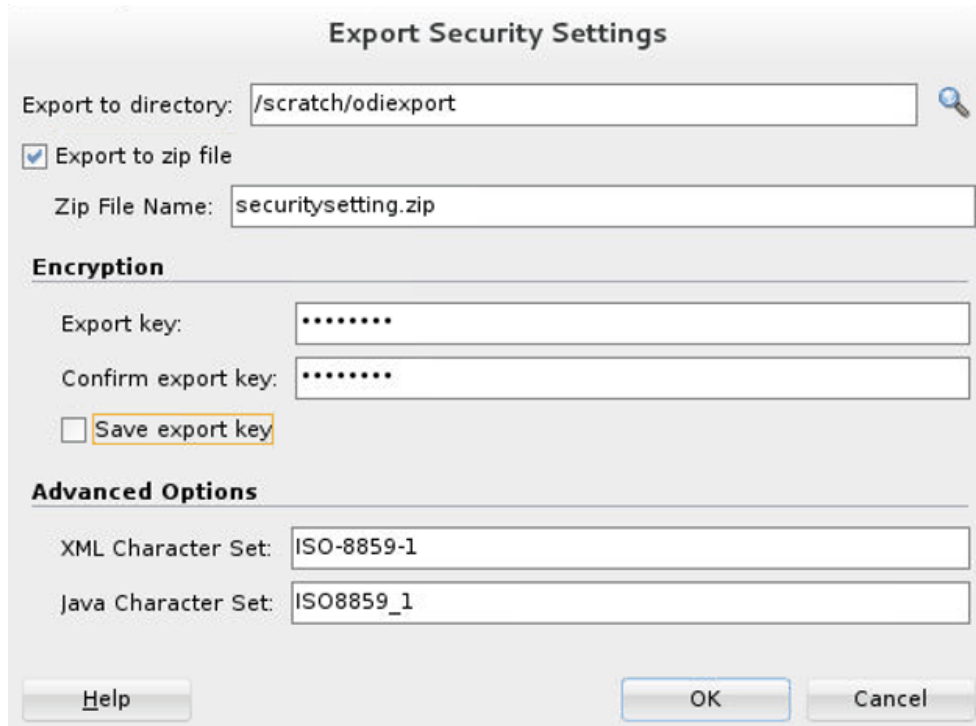
1. Create a local directory path for export as a precaution because the Import process that you run later may delete all the files and sub folders that are inside the import folder that you specify during the Import process.
2. Launch the ODI Studio client and connect to the original ODI repository.
3. Navigate to the Topology tab. From the Connect Navigator (the Topology icon drop-down on the top right side of the navigator pane), select **Export**. As part of the procedures described below, you export files to a local directory.



4. Launch the Export Security Settings wizard from the Export Selection dialog.



5. In Export Security Settings, perform the following and click **OK**:
 - a. Specify the path for the newly created local directory for export in **Export to directory**.
 - b. Select the **Export to zip file** check box.
 - c. Provide the **Zip File Name**.
 - d. Provide the **Export key** to export cipher data (passwords of connections).



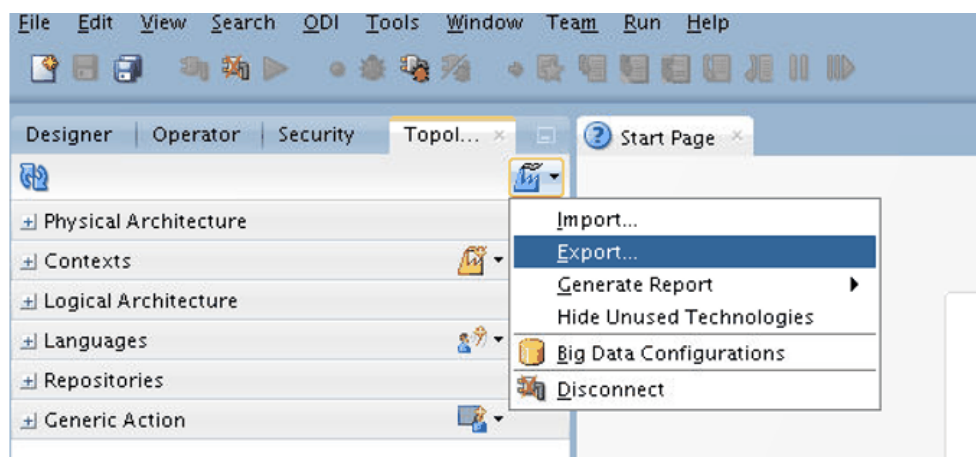
You use this zip export file to import the security settings in a later step.

6. Create a copy of the export directory along with the export zip file for backup.

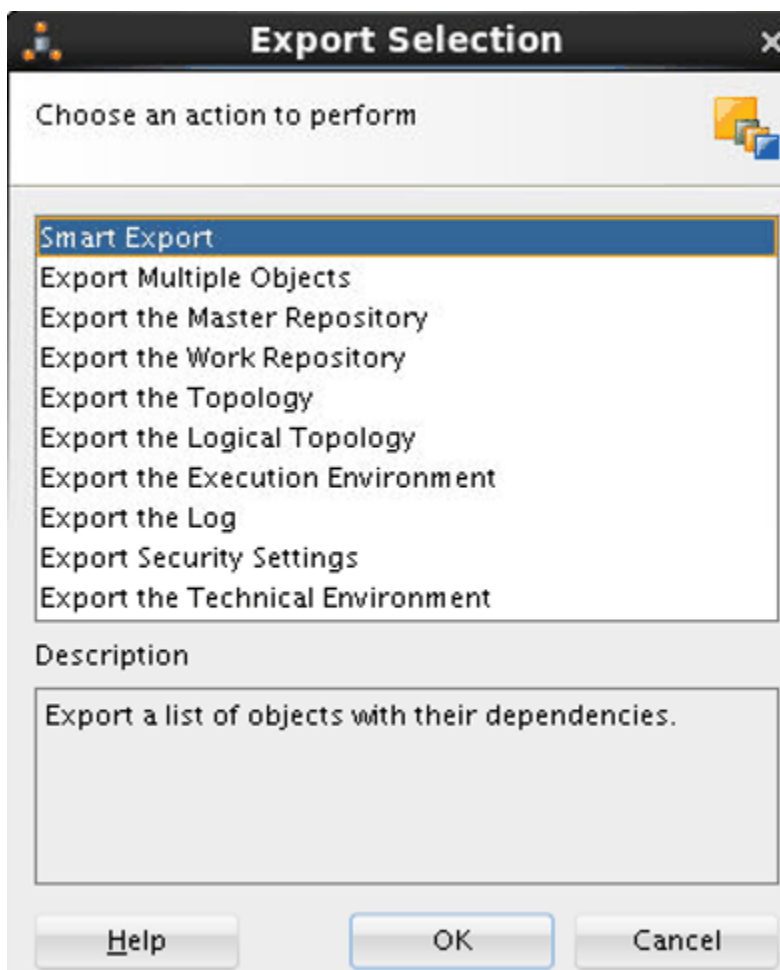
Exporting Topology Settings from Original ODI Repository

You need to export the topology settings from the Original ODI Repository that were created using the Oracle Business Analytics Suite RCU.

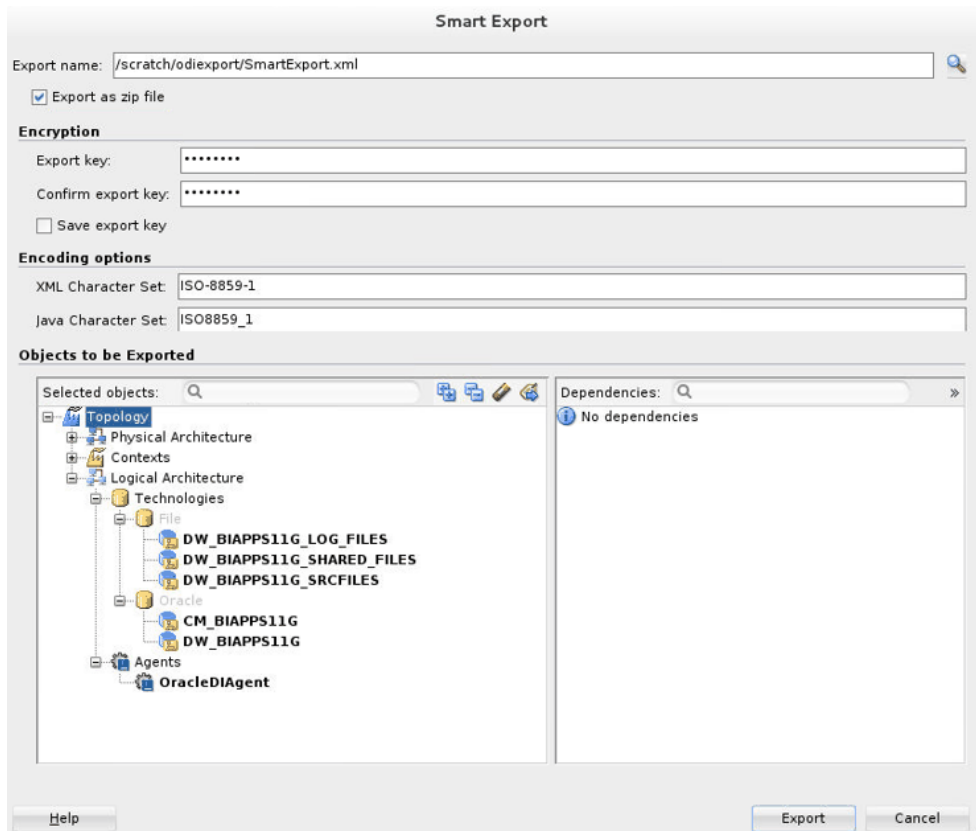
1. Create a local directory path for export as a precaution because the Import process that you run later may delete all the files and sub folders that are inside the import folder that you specify during the Import process.
2. Launch the ODI Studio client and connect to the original ODI repository.
3. Navigate to the Topology tab. From the Connect Navigator (Topology icon drop-down on the top right side of the navigator pane), select **Export**. As part of the procedures described below, you export files to a local directory.



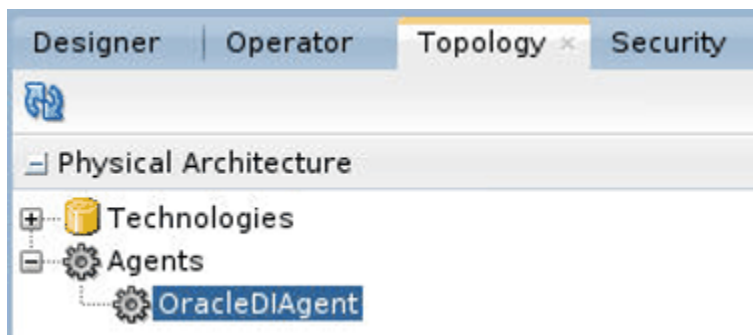
4. Launch the Smart Export wizard from the Export Selection dialog.



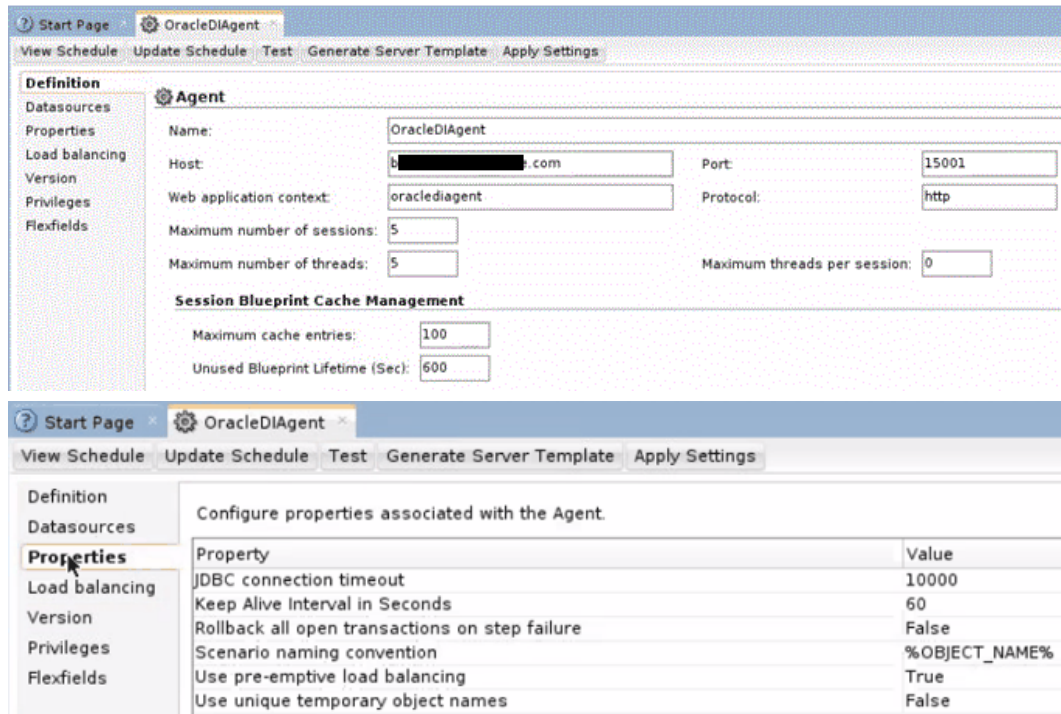
5. In Smart Export, perform the following:
 - a. Specify the newly created local directory for export and provide a meaningful name for the export file.
 - b. Check the **Export as zip file** check box.
 - c. Provide the **Export key** to export along Cipher Data (Connection passwords).
 - d. Select these logical schema objects and logical agents in **Objects to be Exported**:
 - From Files
 - DW_BIAPPS11G_LOG_FILES
 - DW_BIAPPS11G_SHARED_FILES
 - DW_BIAPPS11G_SRCFILES
 - From Oracle
 - CM_BIAPPS11G
 - DW_BIAPPS11G
 - From Agents
 - OracleDIAgent
 - e. Click **Export**.



6. In ODI Studio, click **Topology**, select **Physical Architecture**, and then expand **Agents**.



7. Right-click on the configured agent and then click **Open**.



- Take a screen shot of the properties of the current physical agent to validate it after import.

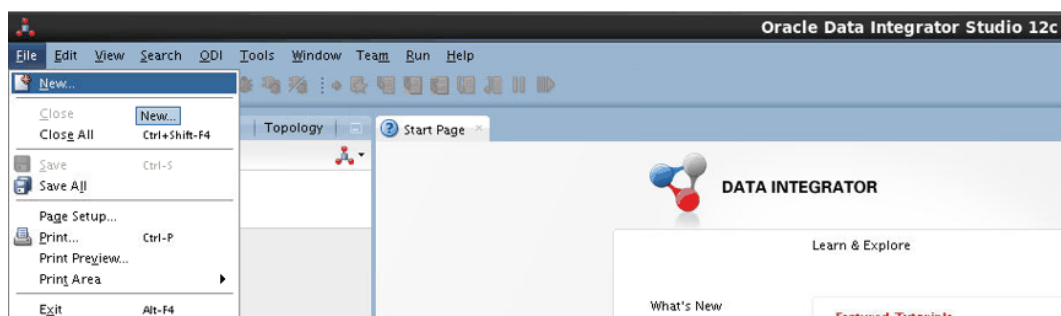
This exports the logical and physical topology including assigned data source num ID values, database connect details, and logical and physical agents.

- Create a copy of the export directory along with the export zip file for backup.

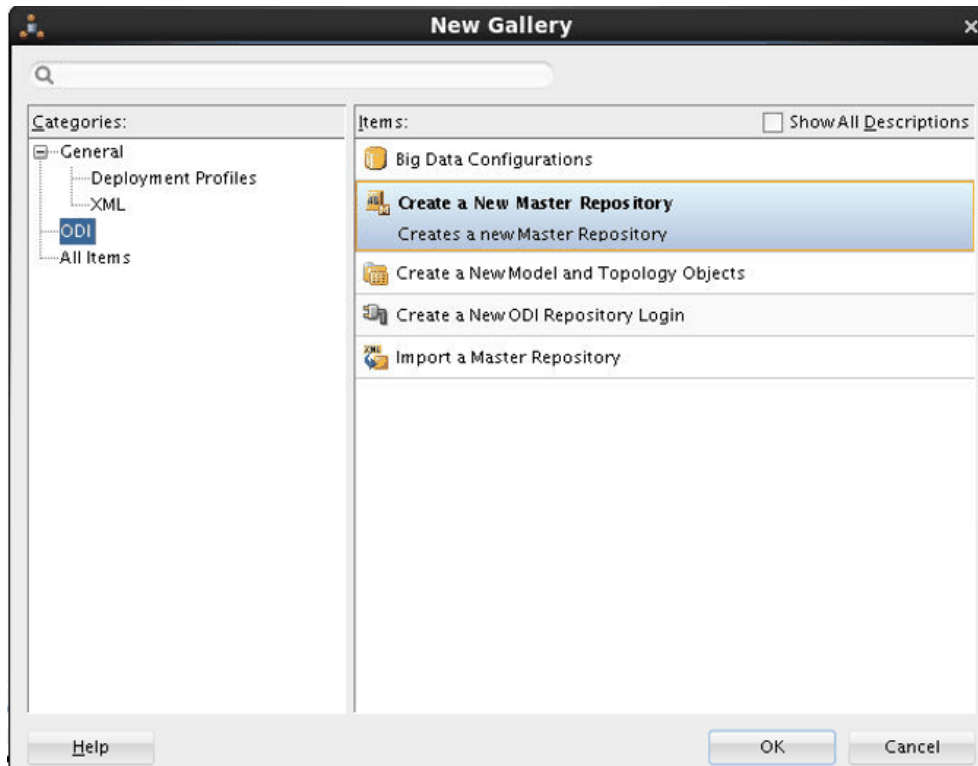
Creating a New Master Repository

Use ODI Studio to create a new master repository.

- Launch ODI Studio.
- Open the New Gallery by selecting **File** and then **New**.



- In the Categories tree of the New Gallery, select **ODI**.
- From the Items list, select **Create a New Master Repository**, and click **OK**.
The Master Repository Creation Wizard opens.

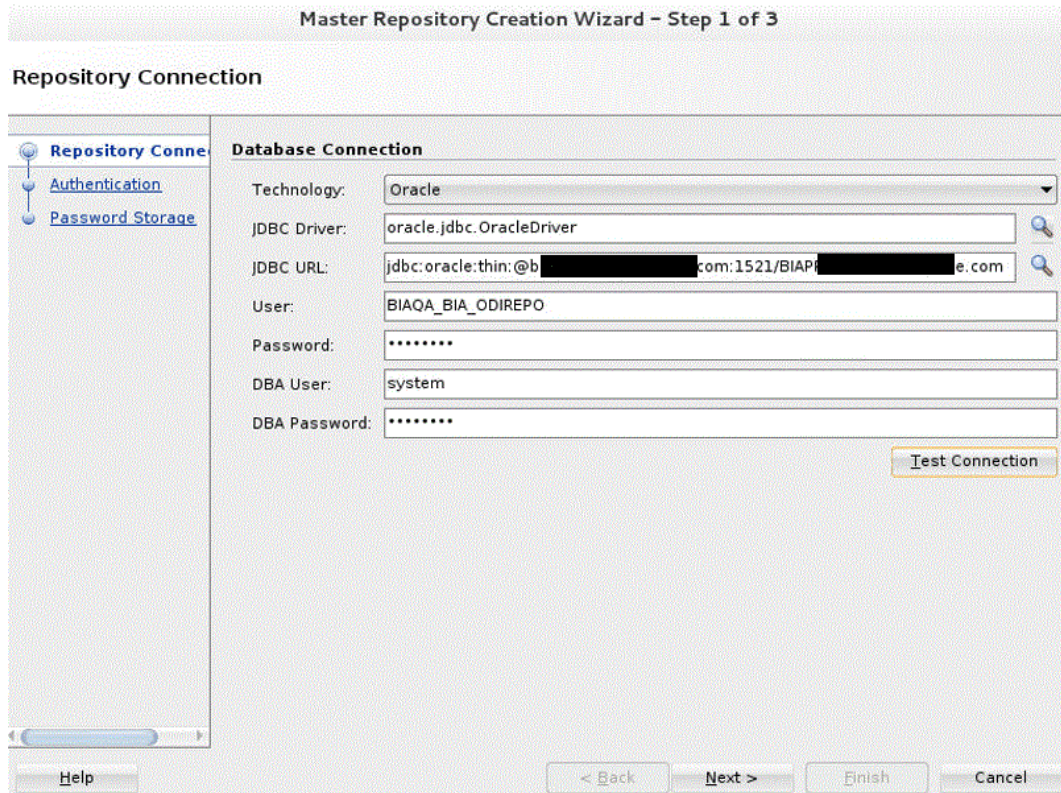


5. Specify the Database Connection properties:

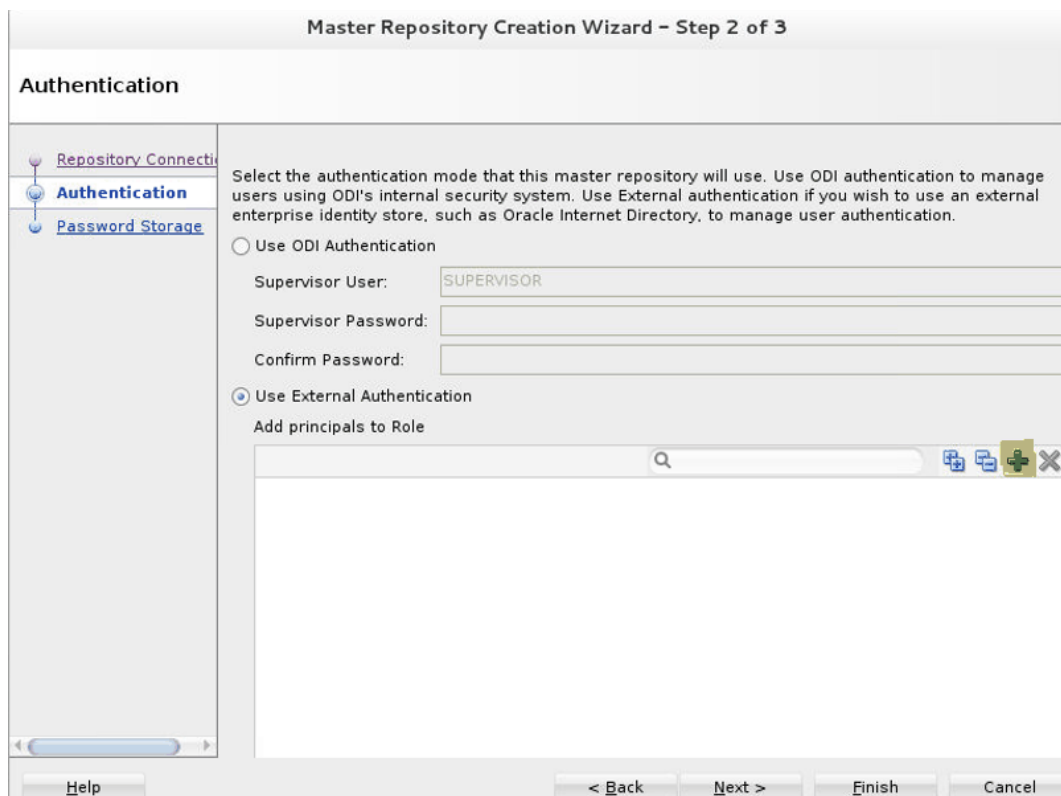
Property	Description
Technology	Select the technology that hosts your Master Repository.
JDBC Driver	The driver used to access the technology that hosts the repository.
JDBC URL	The URL used to establish the JDBC connection to the database. Note that the JDBC Driver and URL properties are synchronized and the default values are technology dependent.
User	The name of the schema where the Master Repository is stored.*
Password	The password for the schema.
DBA User	The user ID on the database with DBA privileges such as SYSTEM. Don't use SYS, which requires logging in as SYSDBA.
DBA Password	The password for the DBA User.

 **Note:**

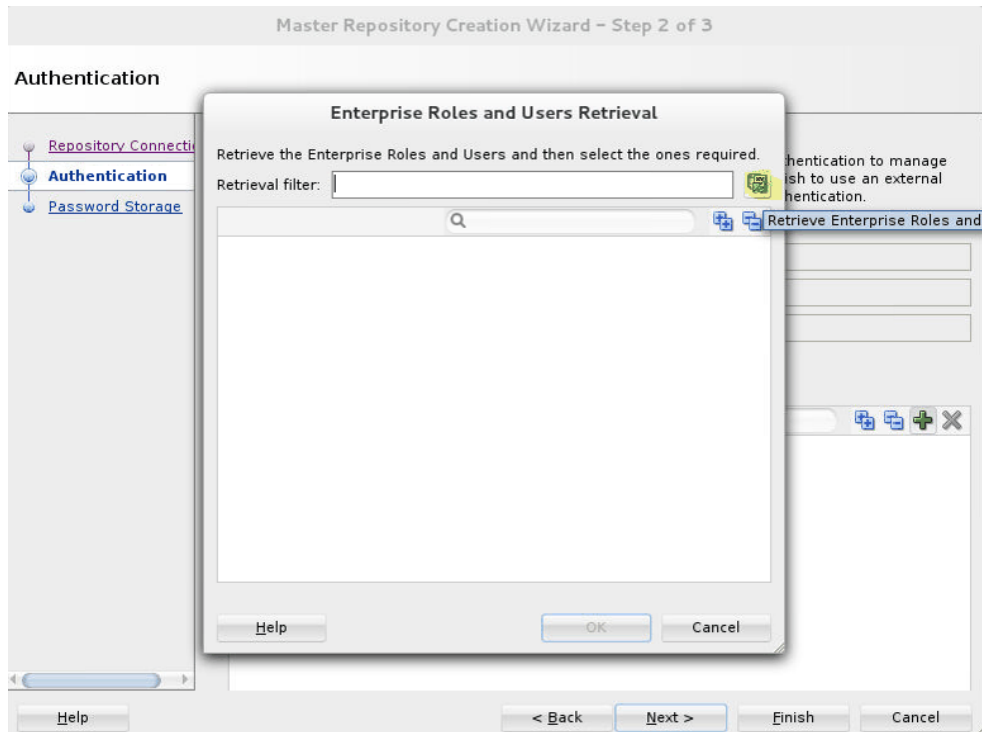
*Use the same credentials used by the installer to create the original repository.



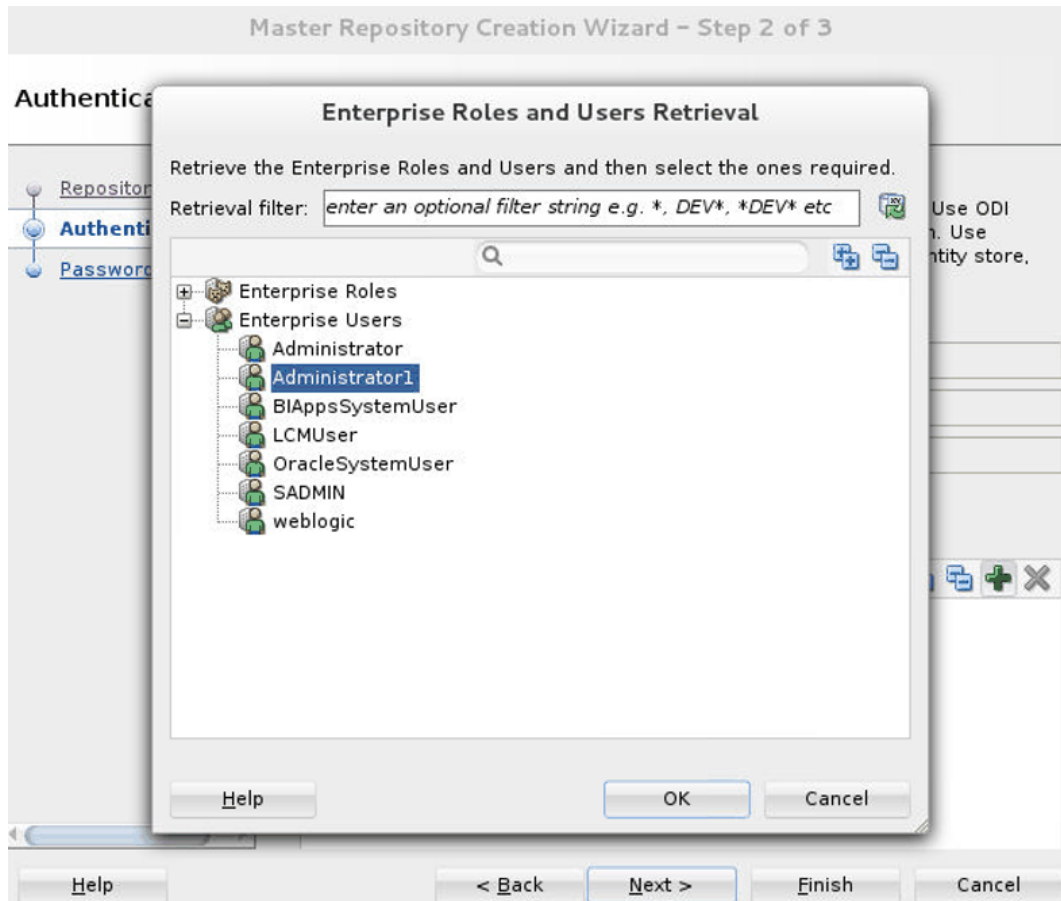
6. (Optional) Click **Test Connection** to test the connection to the Master Repository, then click **Next**.
7. In the Authentication screen, select **Use External Authentication**.



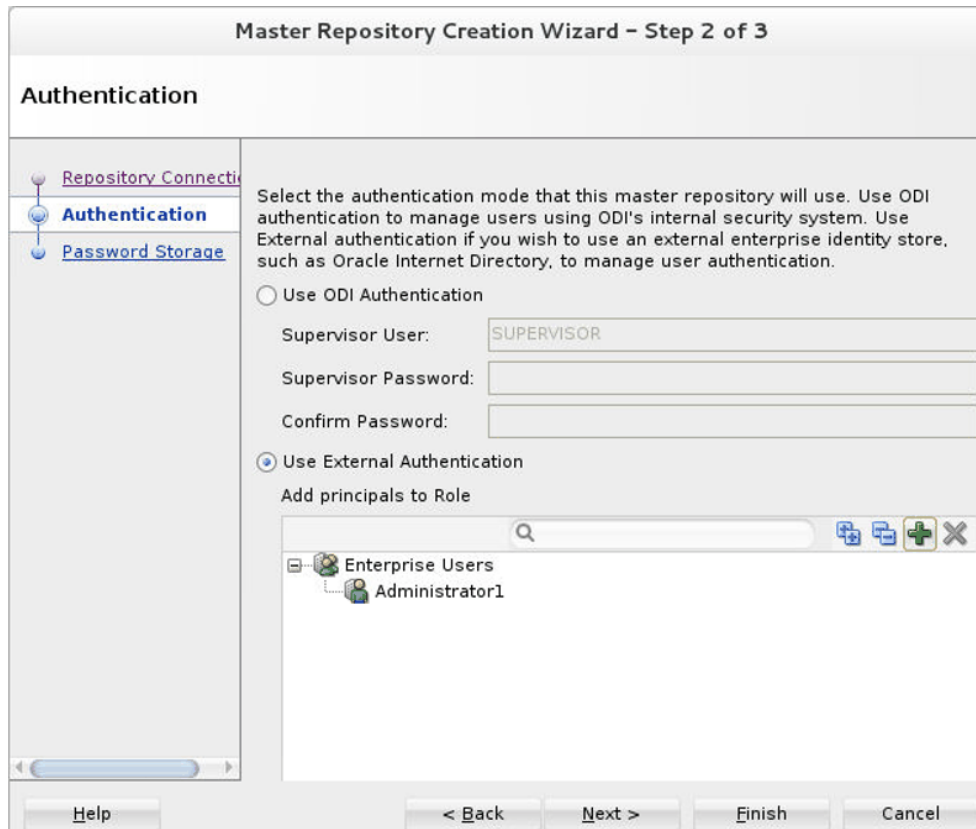
8. Under **Use External Authentication**, click the Add icon +, then in the Enterprise Roles and Users Retrieval dialog, click the Retrieve Enterprise Roles and Users icon.



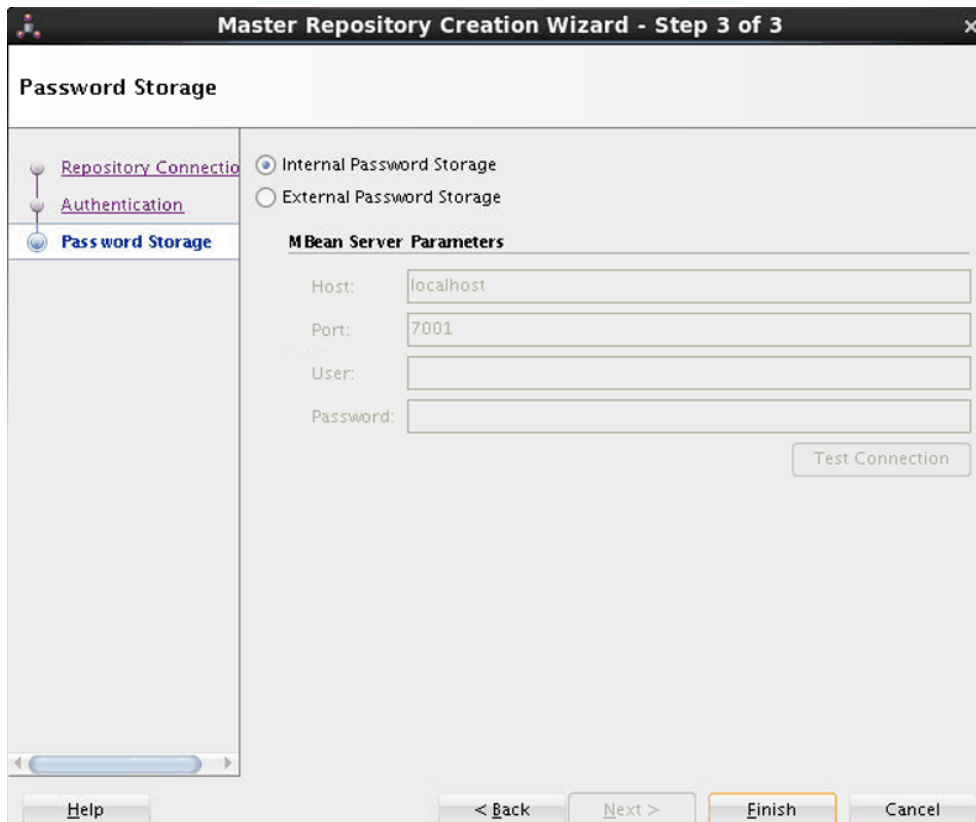
9. In the Enterprise Roles and Users Retrieval dialog, expand **Enterprise Users** and select the users for which Supervisor Role is needed, and then click **OK**.



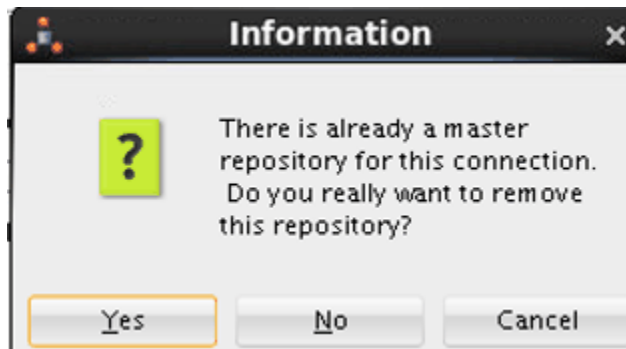
10. Click **Next**.



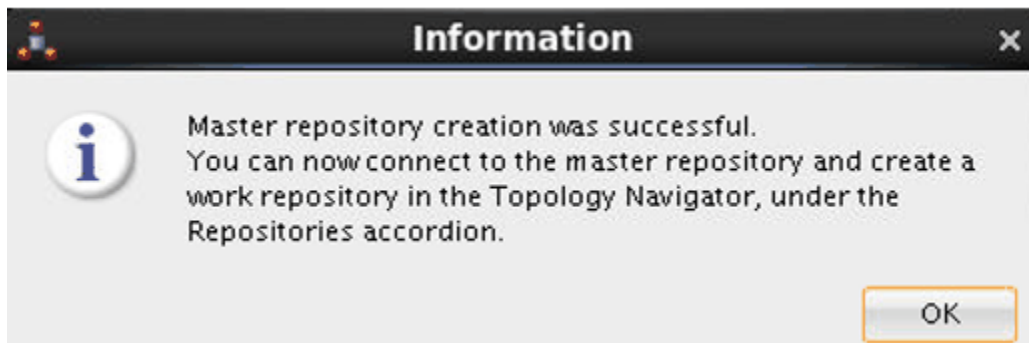
11. In the Password Storage dialog, select **Internal Password Storage**, and then click **Finish**.



- Click **Yes** to indicate you want to remove the existing repository.



- In the information message, click **OK**.



Connecting to the New Master Repository

In this procedure, you create a new Master Repository login connection.

- Open the New Gallery by selecting **File** and then **New**.
- In the Categories tree of the New Gallery, select **ODI**.
- From the Items list, select **Create a New ODI Repository Login**.
- Click **OK**.

The Repository Connection Information dialog opens.

- Specify the ODI connection details:

Property	Description
Login name	A connection alias.
User	The ODI Supervisor user name that was defined during the Oracle BI Applications installation. Note that the Supervisor user name is case sensitive and must be entered exactly as previously defined.
Password	The ODI Supervisor password that was defined during the Oracle BI Applications installation.

- Specify the Database Connection (Master Repository) details:

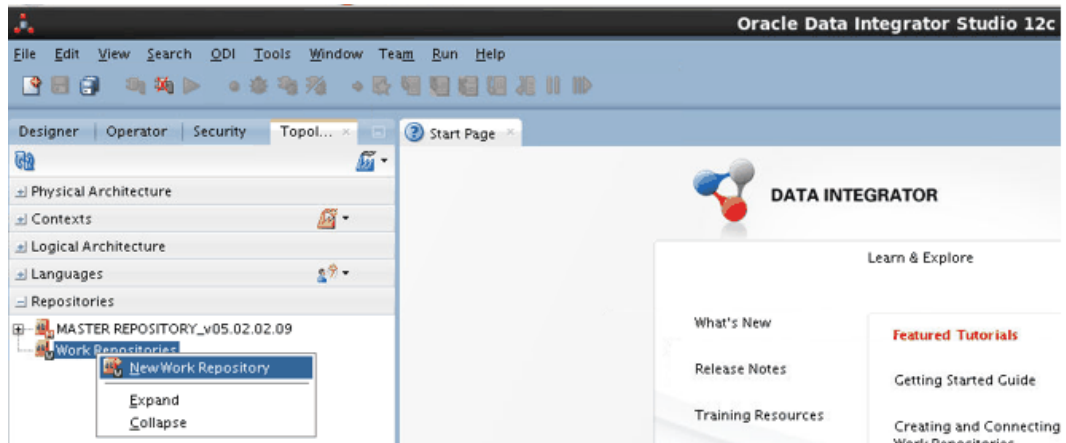
Property	Description
User	The database user login of the schema that contains the Master Repository.
Password	The User password.
Driver List	Select the driver required to connect to the database supporting the Master Repository.
Driver Name	The complete driver name.
URL	The URL used to establish the JDBC connection to the database hosting the repository.

- In the Work Repository area, select **Master Repository Only**.
- (Optional) Click **Test** to verify that the connection is working.
- Click **OK**.

Creating a New Work Repository

Use this procedure to create a new Work Repository.

- In the Topology Navigator, go to the Repositories panel.
- Right-click the **Work Repositories** node, and select **New Work Repository**.



The Create Work Repository Wizard opens.

3. Specify the ODI Work Repository connection details.

The Work Repository needs to be in the same schema as the Master Repository. Provide the same credentials used to create the Master Repository.

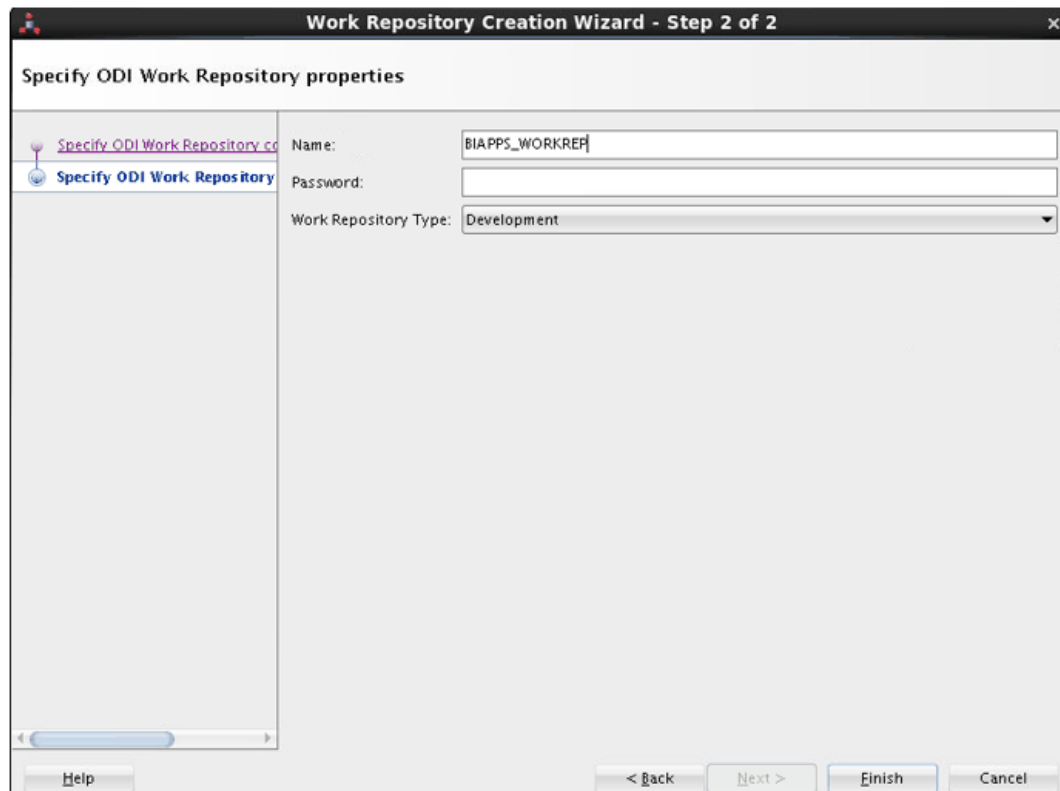
Property	Description
Technology	Select the technology that hosts your Work Repository.
JDBC Driver	The driver used to access the technology that hosts the repository.
JDBC URL	The complete path of the data server to host the Work Repository. Note that the JDBC Driver and URL properties are synchronized and the default values are technology-dependent.
User	The name of the schema where the Work Repository is stored.
Password	The password for the schema.

- (Optional) Click **Test Connection** to verify the connection is working, then click **Next**.

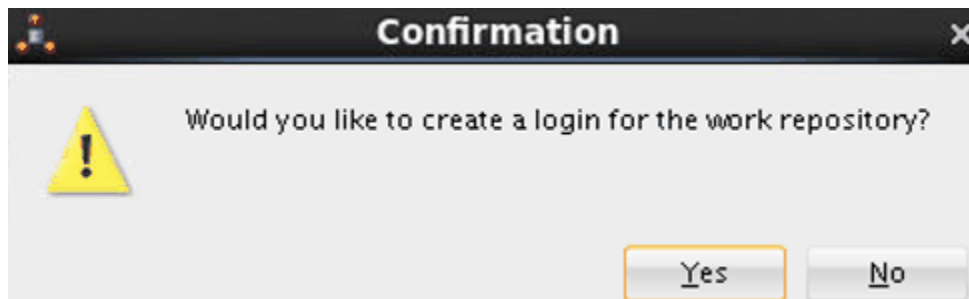
An informational dialog notifies you that a Work Repository already exists.

- Click **No** to overwrite the existing repository.
- Specify the Work Repository properties:

Property	Description
Name	A name for the Work Repository.
Password	This field is optional. The value can be any value. It doesn't correspond to the ODI Supervisor or database passwords.
Work Repository Type	Select Development .



7. Click **Finish**.



8. When the Confirmation dialog asks if you'd like to create a login for the Work Repository, click **No**.
9. Edit the repository connection details to connect to the Work Repository:
 - a. Log off of the Work Repository.
 - b. Open the Repository Connection Information dialog.
 - c. In the Work Repository area, select the **Work Repository** option, and then select the name of the Work Repository you just created.

- d. Click **OK**.

Importing Source Adaptor Metadata Content into the New ODI Repository

In this procedure, you import source-specific metadata content into the new ODI Repository by running the command line utility.

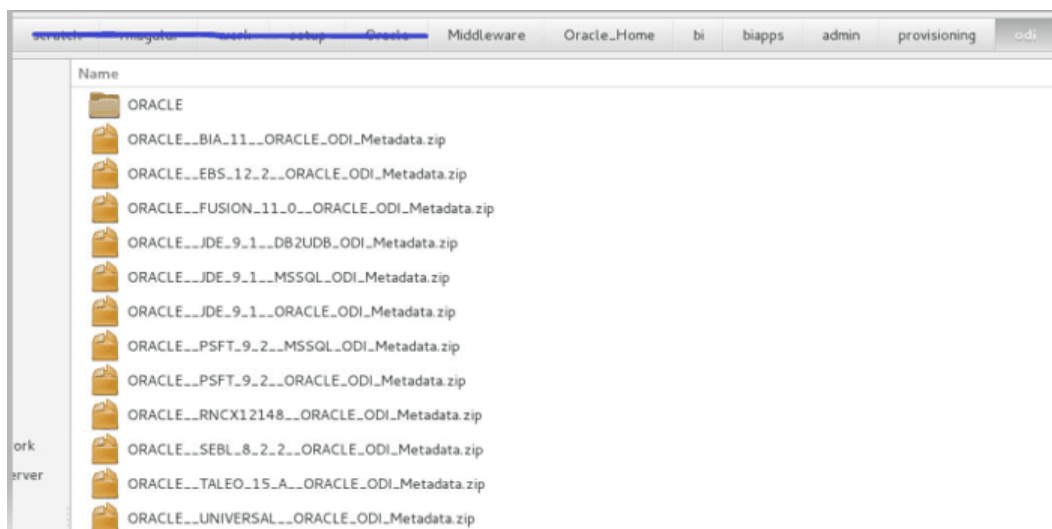
- [Preparing to Import Source Adaptor Metadata Content](#)
- [Running the Command Line Utility to Import Source Adaptor Metadata Content](#)

Preparing to Import Source Adaptor Metadata Content

Use these instructions to import source-specific metadata content into the new ODI Repository.

1. Locate the directory `<ORACLE_HOME>/bi/biapps/admin/provisioning/odi`.

Oracle Home example: `/scratch/product/Oracle/Middleware/Oracle_Home`. The `.zip` files in this directory contain the source-specific metadata content that you import into the new repository. The naming convention for the `.zip` files is `<Source adapter>_<database technology>_ODI_Metadata.zip`.

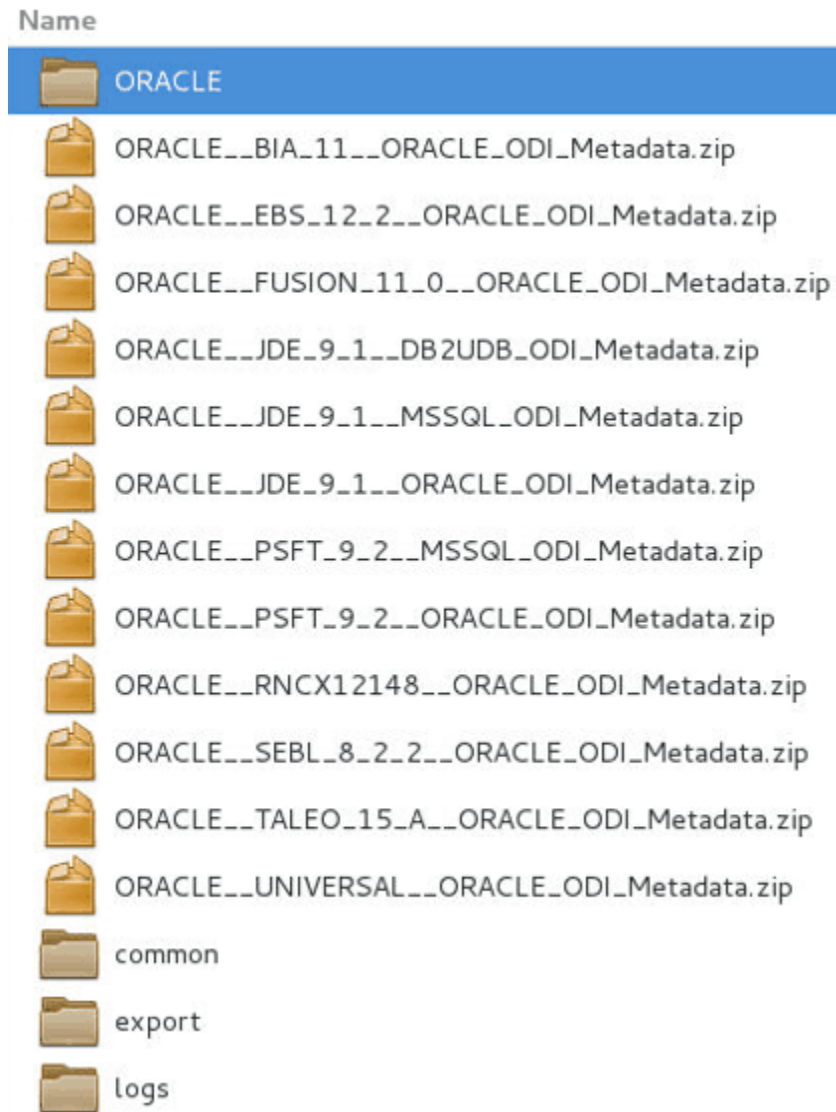


2. Unzip `ORACLE__BIA_11__ORACLE_ODI_Metadata.zip` first and then unzip the appropriate metadata content file for your source adaptor and database technology.

 **Note:**

While unzipping files, don't unzip into a separate folder. Use the Extract here option from the unzipping tools to unzip all metadata zip files.

For example, once you unzip files, you see the following folder structure:



Inside ORACLE folder, you see the following folders:



3. Locate the file `bia-odi-util.jar`, which is stored in `<ORACLE_HOME>/bi/biapps/biaodiutil/lib`.
4. Configure Oracle Data Integrator Studio, if you haven't configured it yet, and ensure that you can sign in.

See [Configure Oracle Data Integrator Studio](#).

5. Ensure that the `jps-config-jse.xml`, `system-jazn-data.xml`, and `cwallet.sso` files have been copied under `<ORACLE_HOME>/odi/studio/bin` while configuring Oracle Data Integrator Studio.

 **Note:**

In the command, the `jps-config-jse.xml` file under `<ORACLE_HOME>/odi/studio/bin` is referred to as `JPSCONFIGFILE`.

6. Create a directory to store the logs of the import process.

In the command, this directory is referred to as `LOGDIR`.

7. Obtain these ODI Repository connection details:

- JDBC URL for the ODI Repository database
- JDBC driver
- Work Repository name
- ODI Repository database user name and password
- ODI user and password

This connection information is displayed in ODI Studio in the Repository Connection Details dialog.

Running the Command Line Utility to Import Source Adaptor Metadata Content

Use this procedure to import the metadata content files into the new ODI Repository. You first import the metadata for Oracle BI Applications, after which you import the metadata for the appropriate source adaptors.

For potential performance enhancement, Oracle recommends running the command line utility on the database server machine.

To import metadata content by running the command line utility:

1. Open a command line utility.
2. Navigate to the `<ORACLE_HOME>/odi/studio/bin` directory.
3. Run the following command to import the metadata of Oracle BI Applications first, which is mandatory before importing any source-specific metadata content.

The PLV code for this adaptor is `BIA_11`.

On UNIX/Linux:

```
<JAVA_HOME>/bin/java -client -Dfile.encoding=UTF-8 -classpath
  "<ORACLE_HOME>/odi/sdk/lib/*:<ORACLE_HOME>/oracle_common/jlib/
  *:<ORACLE_HOME>/oracle_common/modules/oracle.idm/*:<ORACLE_HOME>/
  oracle_common/modules/oracle.jps/*:<ORACLE_HOME>/oracle_common/modules/
  oracle.jdbc/*:<ORACLE_HOME>/oracle_common/modules/oracle.nlsrtl/
  *:<ORACLE_HOME>/oracle_common/modules/oracle.ucp.jar:<ORACLE_HOME>/bi/
```

```

biapps/lib/*:<ORACLE_HOME>/bi/biapps/biaodiutil/lib/bia-odi-
util.jar"
-Dcommon.components.home=<ORACLE_HOME>/oracle_common/
oracle.apps.biaodiutil.Import
MODE=IMPORT FILEDIR=<EXP_FILE_DIR> PLV=<PLV_CODE>
JPSCONFIGFILE=<JPS_CONFIG_FILE>
LOGDIR=<LOG_DIR>

```

For example:

```

/scratch/jdk/jdk_1.8.0_201/bin/java -client -Dfile.encoding=UTF-8 -
classpath
"/scratch/product/Oracle/Middleware/
Oracle_Home/odi/sdk/lib/*:/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/jlib/*:/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/modules/oracle.idm/*:/scratch/product/
Oracle/Middleware/Oracle_Home/oracle_common/modules/oracle.jps/*:/
scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/modules/
oracle.jdbc/*:/scratch/product/Oracle/Middleware/Oracle_Home/
oracle_common/modules/oracle.nlsrtl/*:/scratch/product/Oracle/
Middleware/Oracle_Home/oracle_common/modules/oracle.ucp.jar:/
scratch/product/Oracle/Middleware/Oracle_Home/bi/biapps/lib/*:/
scratch/product/Oracle/Middleware/Oracle_Home/bi/biapps/
biaodiutil/lib/bia-odi-util.jar"
-Dcommon.components.home=/scratch/product/Oracle/
Middleware/Oracle_Home/oracle_common/
oracle.apps.biaodiutil.Import MODE=IMPORT
FILEDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/
biapps/admin/provisioning/odi/
PLV=BIA_11
JPSCONFIGFILE=/scratch/product/Oracle/Middleware/
Oracle_Home/odi/studio/bin/jps-config-jse.xml
LOGDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/
biapps/admin/provisioning/odi/logs/

```

On Windows:

```

<JAVA_HOME>\bin\java -client -Dfile.encoding=UTF-8 -classpath
"<ORACLE_HOME>\odi\sdk\lib\*;<ORACLE_HOME>\oracle_common\jlib\*;<ORA
CLE_HOME>\oracle_common\modules\oracle.idm\*;<ORACLE_HOME>\oracle_co
mmon\modules\oracle.jps\*;<ORACLE_HOME>\oracle_common\modules\oracle
.jdbc\*;<ORACLE_HOME>\oracle_common\modules\oracle.nlsrtl\*;<ORACLE_
HOME>\oracle_common\modules\oracle.ucp.jar;<ORACLE_HOME>\bi\biapps\l
ib\*;<ORACLE_HOME>\bi\biapps\biaodiutil\lib\bia-odi-util.jar"
-Dcommon.components.home=<ORACLE_HOME>\oracle_common\
oracle.apps.biaodiutil.Import
MODE=IMPORT FILEDIR=<EXP_FILE_DIR> PLV=<PLV_CODE>
JPSCONFIGFILE=<JPS_CONFIG_FILE>
LOGDIR=<LOG_DIR>

```

For example:

```
C:\jdk\jdk_1.8.0_201\bin\java -client -Dfile.encoding=UTF-8 -classpath
"C:\product\Oracle\Middleware\Oracle_Home\odi\sdk\lib\*;C:\product\Oracle\
Middleware\Oracle_Home\oracle_common\jlib\*;C:\product\Oracle\Middleware\O
racle_Home\oracle_common\modules\oracle.idm\*;C:\product\Oracle\Middleware
\Oracle_Home\oracle_common\modules\oracle.jps\*;C:\product\Oracle\Midde
lware\Oracle_Home\oracle_common\modules\oracle.jdbc\*;C:\product\Oracle\Midde
lware\Oracle_Home\oracle_common\modules\oracle.nlsrtl\*;C:\product\Oracle\
Middleware\Oracle_Home\oracle_common\modules\oracle.ucp.jar;C:\product\Ora
cle\Middleware\Oracle_Home\bi\biapps\lib\*;C:\product\Oracle\Middleware\Ora
cle_Home\bi\biapps\biaodiutil\lib\bia-odi-util.jar"
-
Dcommon.components.home=C:\product\Oracle\Middleware\Oracle_Home\oracle_co
mmon\
    oracle.apps.biaodiutil.Import MODE=IMPORT

FILEDIR=C:\product\Oracle\Middleware\Oracle_Home\bi\biapps\admin\provision
ing\odi\
    PLV=BIA_11

JPSCONFIGFILE=C:\product\Oracle\Middleware\Oracle_Home\odi\studio\bin\jps-
config-jse.xml

LOGDIR=C:\product\Oracle\Middleware\Oracle_Home\bi\biapps\admin\provisioni
ng\odi\logs\
```

4. Run the following command to import the source-specific metadata content. The PLV code and SRCTECH for the source-specific metadata content are given in the following parameters table.

This table describes the parameters:

Parameter	Description
MODE	(Mandatory) Possible values: <ul style="list-style-type: none"> • IMPREGEN - Use this mode to import the adaptor metadata content and generate scenarios for that adaptor. • IMPORT — Use this mode to import metadata content only, without generating scenarios. A parameter file is generated that allows you to do the scenario generation at a later time. • SCEREGEN — Use this mode to generate scenarios only. Before running this mode, you need to have imported content using the IMPORT mode so that the parameter file required for scenario generation is already present.
FILEDIR	(Mandatory) The complete path for the export file directory files.

Parameter	Description
CONNFILE	<p>(Optional) The complete path of the file containing the Connection Details. If this parameter isn't used, the utility prompts for the connection details.</p> <p>An example of a connection file is as follows:</p> <pre># ===== # ODI Connection Information # ===== # Connection Information for Destination Oracle Data #Integrator Repository MasterJDBCdriver=oracle.jdbc.OracleDriver MasterJDBCurl=jdbc:oracle:thin:@<host>:1521:orcl MasterJDBCUser=ODI_REP MasterJDBCPass=odirep WorkName=BIAPPS_WORKREP ODIUser=BIAppsAdminUser ODIPass=odipass</pre>
LOGLEVEL	<p>(Optional) Possible values are 1, 3, or 5. The default value is 1, with 5 being the most verbose.</p>
LOGDIR	<p>The complete path for the log directory.</p> <p>The following log files are created:</p> <ul style="list-style-type: none"> • Summary — Contains high level log information. The naming convention is Summary_ImportUtility_<DATE_TIME><ADAPTOR><SRCTECH>_log. • Import — Contains details about the import. Has a log and error file. The naming convention is BasicImport_<DATE_TIME>_<ADAPTOR>_log. • Scenario Generation — Contains details about the scenario generation. Has a log and error file. The naming convention is RegenScen_<DATE_TIME>_<ADAPTOR_MAPPING_FOLDER>_log.
PLV	<p>(Mandatory) The PLV code for the adaptor being imported.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • EBS_12_2 • FUSION_11_0 • JDE_9_1 • PSFT_9_2 • SEBL_8_2_2 • RNCX12148 • TALEO_15_A • UNIVERSAL
SRCTECH	<p>(Mandatory) The source technology for the adaptor being imported.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • For IBM DB2 UDB: IBM_DB2_UDB • For SQL Server: MICROSOFT_SQL_SERVER • For ORACLE: ORACLE • For Oracle Fusion (via BI Server): ORACLE_BI • For UNIVERSAL: FILE <p>If you've multiple sources on both Oracle and non-Oracle databases, you need to import the PLV metadata for the Oracle database using ORACLE (or ORACLE_BI for Oracle Fusion sources) as the SRCTECH value.</p>

Parameter	Description
JPSCONFIGFILE	(Mandatory if the repository has already been enabled for external authentication.) The path and name of the JPSCONFIGFILE.
SSOFILE	(Optional) The path and name of the SSO file.

On Linux:

For example:

```
/scratch/jdk/jdk_1.8.0_201/bin/java -client -Dfile.encoding=UTF-8 -
classpath
    "/scratch/product/Oracle/Middleware/Oracle_Home/odi/sdk/lib/*:/
scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/jlib/*:/
scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/modules/
oracle.idm/*:/scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/
modules/oracle.jps/*:/scratch/product/Oracle/Middleware/Oracle_Home/
oracle_common/modules/oracle.jdbc/*:/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/modules/oracle.nlsrtl/*:/scratch/product/Oracle/
Middleware/Oracle_Home/oracle_common/modules/oracle.ucp.jar:/scratch/
product/Oracle/Middleware/Oracle_Home/bi/biapps/lib/*:/scratch/product/
Oracle/Middleware/Oracle_Home/bi/biapps/biaodiutil/lib/bia-odi-util.jar"
    -Dcommon.components.home=/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/
    oracle.apps.biaodiutil.Import MODE=IMPORT
    FILEDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/
biapps/admin/provisioning/odi/
    PLV=JDE_9_1 SRCTECH=MICROSOFT_SQL_SERVER
    JPSCONFIGFILE=/scratch/product/Oracle/Middleware/
Oracle_Home/odi/studio/bin/jps-config-jse.xml
    LOGDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/biapps/
admin/provisioning/odi/logs/
```

Repeat Step 4 if you want to import any other source specific metadata content by choosing the appropriate PLV code and SRCTECH parameter.

Following are few sample commands for importing other source specific metadata:

JDE_9_1 for MSSQL:

```
/scratch/jdk/jdk_1.8.0_201/bin/java -client -Dfile.encoding=UTF-8 -
classpath
    "/scratch/product/Oracle/Middleware/Oracle_Home/odi/sdk/lib/*:/
scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/jlib/*:/
scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/modules/
oracle.idm/*:/scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/
modules/oracle.jps/*:/scratch/product/Oracle/Middleware/Oracle_Home/
oracle_common/modules/oracle.jdbc/*:/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/modules/oracle.nlsrtl/*:/scratch/product/Oracle/
Middleware/Oracle_Home/oracle_common/modules/oracle.ucp.jar:/scratch/
product/Oracle/Middleware/Oracle_Home/bi/biapps/lib/*:/scratch/product/
Oracle/Middleware/Oracle_Home/bi/biapps/biaodiutil/lib/bia-odi-util.jar"
    -Dcommon.components.home=/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/
    oracle.apps.biaodiutil.Import MODE=IMPORT
    FILEDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/
```

```

biapps/admin/provisioning/odi/
    PLV=JDE_9_1 SRCTECH=MICROSOFT_SQL_SERVER
    JPSCONFIGFILE=/scratch/product/Oracle/Middleware/
Oracle_Home/odi/studio/bin/jps-config-jse.xml
    LOGDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/
biapps/admin/provisioning/odi/logs/

```

JDE_9_1 for ORACLE:

```

/scratch/jdk/jdk_1.8.0_201/bin/java -client -Dfile.encoding=UTF-8 -
classpath
    "/scratch/product/Oracle/Middleware/
Oracle_Home/odi/sdk/lib/*:/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/jlib/*:/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/modules/oracle.idm/*:/scratch/product/
Oracle/Middleware/Oracle_Home/oracle_common/modules/oracle.jps/*:/
scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/modules/
oracle.jdbc/*:/scratch/product/Oracle/Middleware/Oracle_Home/
oracle_common/modules/oracle.nlsrtl/*:/scratch/product/Oracle/
Middleware/Oracle_Home/oracle_common/modules/oracle.ucp.jar:/
scratch/product/Oracle/Middleware/Oracle_Home/bi/biapps/lib/*:/
scratch/product/Oracle/Middleware/Oracle_Home/bi/biapps/
biaodiutil/lib/bia-odi-util.jar"
    -Dcommon.components.home=/scratch/product/Oracle/
Middleware/Oracle_Home/oracle_common/
    oracle.apps.biaodiutil.Import MODE=IMPORT
    FILEDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/
biapps/admin/provisioning/odi/
    PLV=JDE_9_1 SRCTECH=ORACLE
    JPSCONFIGFILE=/scratch/product/Oracle/Middleware/
Oracle_Home/odi/studio/bin/jps-config-jse.xml
    LOGDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/
biapps/admin/provisioning/odi/logs/

```

PSFT_9_2 for MSSQL:

```

/scratch/jdk/jdk_1.8.0_201/bin/java -client -Dfile.encoding=UTF-8 -
classpath
    "/scratch/product/Oracle/Middleware/
Oracle_Home/odi/sdk/lib/*:/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/jlib/*:/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/modules/oracle.idm/*:/scratch/product/
Oracle/Middleware/Oracle_Home/oracle_common/modules/oracle.jps/*:/
scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/modules/
oracle.jdbc/*:/scratch/product/Oracle/Middleware/Oracle_Home/
oracle_common/modules/oracle.nlsrtl/*:/scratch/product/Oracle/
Middleware/Oracle_Home/oracle_common/modules/oracle.ucp.jar:/
scratch/product/Oracle/Middleware/Oracle_Home/bi/biapps/lib/*:/
scratch/product/Oracle/Middleware/Oracle_Home/bi/biapps/
biaodiutil/lib/bia-odi-util.jar"
    -Dcommon.components.home=/scratch/product/Oracle/
Middleware/Oracle_Home/oracle_common/
    oracle.apps.biaodiutil.Import MODE=IMPORT
    FILEDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/

```

```

biapps/admin/provisioning/odi/
    PLV=PSFT_9_2 SRCTECH=MICROSOFT_SQL_SERVER
    JPSCONFIGFILE=/scratch/product/Oracle/Middleware/
Oracle_Home/odi/studio/bin/jps-config-jse.xml
    LOGDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/biapps/
admin/provisioning/odi/logs/

```

PSFT_9_2 for ORACLE:

```

/scratch/jdk/jdk_1.8.0_201/bin/java -client -Dfile.encoding=UTF-8 -
classpath
    "/scratch/product/Oracle/Middleware/Oracle_Home/odi/sdk/lib/*:/
scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/jlib/*:/
scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/modules/
oracle.idm/*:/scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/
modules/oracle.jps/*:/scratch/product/Oracle/Middleware/Oracle_Home/
oracle_common/modules/oracle.jdbc/*:/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/modules/oracle.nlstrtl/*:/scratch/product/Oracle/
Middleware/Oracle_Home/oracle_common/modules/oracle.ucp.jar:/scratch/
product/Oracle/Middleware/Oracle_Home/bi/biapps/lib/*:/scratch/product/
Oracle/Middleware/Oracle_Home/bi/biapps/biaodiutil/lib/bia-odi-util.jar"
    -Dcommon.components.home=/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/
    oracle.apps.biaodiutil.Import MODE=IMPORT
    FILEDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/
biapps/admin/provisioning/odi/
    PLV=PSFT_9_2 SRCTECH=ORACLE
    JPSCONFIGFILE=/scratch/product/Oracle/Middleware/
Oracle_Home/odi/studio/bin/jps-config-jse.xml
    LOGDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/biapps/
admin/provisioning/odi/logs/

```

UNIVERSAL ADAPTOR :

```

/scratch/jdk/jdk_1.8.0_201/bin/java -client -Dfile.encoding=UTF-8 -
classpath
    "/scratch/product/Oracle/Middleware/Oracle_Home/odi/sdk/lib/*:/
scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/jlib/*:/
scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/modules/
oracle.idm/*:/scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/
modules/oracle.jps/*:/scratch/product/Oracle/Middleware/Oracle_Home/
oracle_common/modules/oracle.jdbc/*:/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/modules/oracle.nlstrtl/*:/scratch/product/Oracle/
Middleware/Oracle_Home/oracle_common/modules/oracle.ucp.jar:/scratch/
product/Oracle/Middleware/Oracle_Home/bi/biapps/lib/*:/scratch/product/
Oracle/Middleware/Oracle_Home/bi/biapps/biaodiutil/lib/bia-odi-util.jar"
    -Dcommon.components.home=/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/
    oracle.apps.biaodiutil.Import MODE=IMPORT
    FILEDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/
biapps/admin/provisioning/odi/
    PLV=UNIVERSAL SRCTECH=FILE
    JPSCONFIGFILE=/scratch/product/Oracle/Middleware/
Oracle_Home/odi/studio/bin/jps-config-jse.xml

```

```
LOGDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/
biapps/admin/provisioning/odi/logs/
```

JDE_9_1 for IBM_DB2_UDB

```
/scratch/jdk/jdk_1.8.0_201/bin/java -client -Dfile.encoding=UTF-8
-classpath
"/scratch/product/Oracle/Middleware/
Oracle_Home/odi/sdk/lib/*:/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/jlib/*:/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/modules/oracle.idm/*:/scratch/product/
Oracle/Middleware/Oracle_Home/oracle_common/modules/oracle.jps/*:/
scratch/product/Oracle/Middleware/Oracle_Home/oracle_common/modules/
oracle.jdbc/*:/scratch/product/Oracle/Middleware/Oracle_Home/
oracle_common/modules/oracle.nlsrtl/*:/scratch/product/Oracle/
Middleware/Oracle_Home/oracle_common/modules/oracle.ucp.jar:/
scratch/product/Oracle/Middleware/Oracle_Home/bi/biapps/lib/*:/
scratch/product/Oracle/Middleware/Oracle_Home/bi/biapps/
biaodiutil/lib/bia-odi-util.jar"
-Dcommon.components.home=/scratch/product/Oracle/Middleware/
Oracle_Home/oracle_common/
oracle.apps.biaodiutil.Import MODE=IMPORT
FILEDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/
biapps/admin/provisioning/odi/
PLV=JDE_9_1 SRCTECH=IBM_DB2_UDB
JPSCONFIGFILE=/scratch/product/Oracle/Middleware/
Oracle_Home/odi/studio/bin/jps-config-jse.xml
LOGDIR=/scratch/product/Oracle/Middleware/Oracle_Home/bi/
biapps/admin/provisioning/odi/logs/
```

Note:

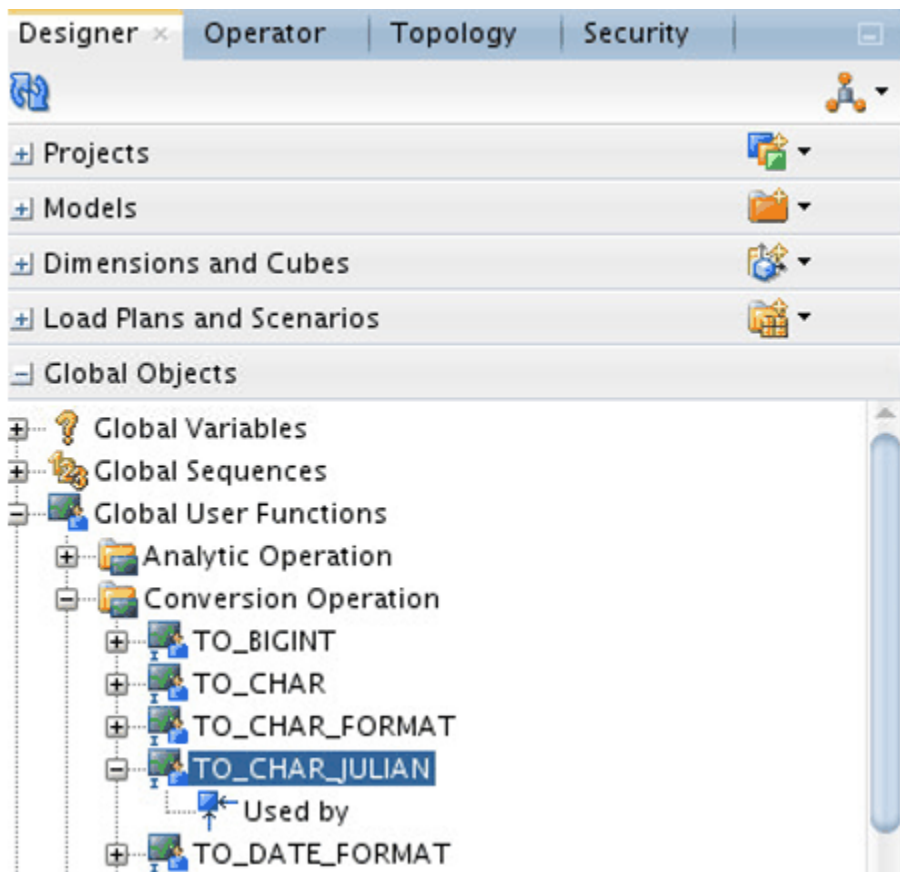
All the adaptor metadata zip files are delivered along with the package scenarios. When you run the command line utility to import these files, ensure that you use import mode as `IMPORT` which would import the adaptor metadata content for that adaptor.

5. Enter the ODI Repository connection details when prompted.
6. Monitor the process by viewing the command line window or the log files being written to the logs directory.

This process can be lengthy and can last from 1 to 10 hours, depending on the number of adaptors for which you are importing metadata content.
7. When the import process is complete, log into ODI Studio and verify the data stores and mappings for the new adaptor are present.
8. Make a copy of file `<LOGDIR>/sceregenparams.properties` as `<LOGDIR>/sceregenparams.properties_<PLV>`.

For example, copy `PLV=BIA_11` in `<LOGDIR>/sceregenparams.properties_BIA_11` immediately after completing the import process for each adaptor.

9. Repeat steps 1 through 7 for each adaptor you need to import, substituting the appropriate PLV code and the source technology.
10. After importing all the adaptors, sign in to ODI Studio.
11. Click Designer, expand Global Objects, click Global User Functions, and then click Conversion Operation.
12. Expand `TO_CHAR_JULIAN` and check for **Used by**. If there are any references, then note them as you must regenerate the scenarios using these references.



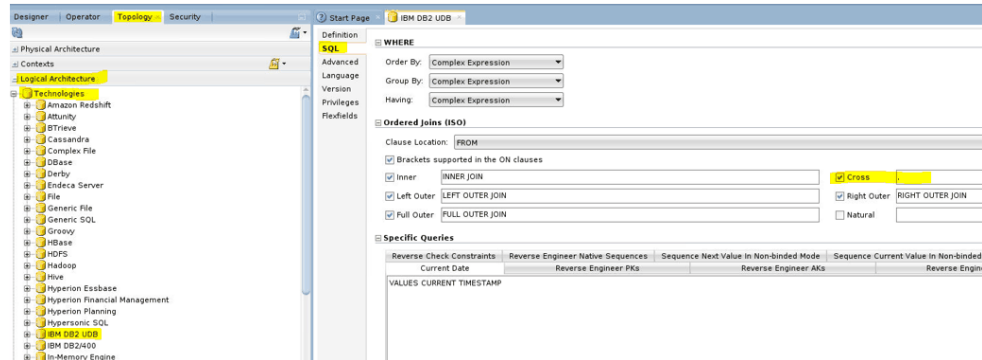
 **Note:**

The user defined functions `TO_CHAR` and `TO_CHAR_JULIAN` have similar implementation syntax `TO_CHAR($VALUE)` and `TO_CHAR($VALUE,'J')`. Sometimes after import, the `TO_CHAR` function references are changed to `TO_CHAR_JULIAN` in the expressions. The `TO_CHAR_JULIAN` user-defined function isn't used in Oracle BI Applications 11.1.1.10.3.

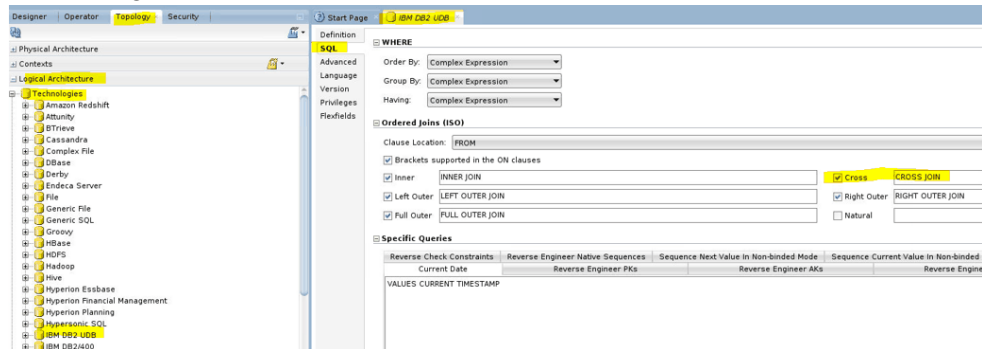
13. Right-click the `TO_CHAR_JULIAN` user-defined function, click **Delete**, and then click **Yes** to remove it.
14. Sign out of ODI Studio.
15. For IBM DB2 UDB Technology, do the following:
 - Run in import mode as `IMPORT` to import the adaptor metadata.
 - After import is completed, sign in to ODI Studio.

- Click on **Topology**, expand **Logical Architecture**, and then expand **Technologies**.
- Right-click **IBM DB2 UDB** and then click **Open**.
- Click **SQL** and in the **Cross** join syntax, and change “,” to **CROSS JOIN**.

Before changes:



After changes:



Click **Save** and sign out of ODI Studio.

16. For each of the imported adaptors, repeat the steps to regenerate scenarios one by one.
17. Overwrite the file `<LOGDIR>/sceregenparams.properties` with backup file `<LOGDIR>/sceregenparams.properties_<PLV>` for the adaptor scenarios to be regenerated using this Unix command: `cp <LOGDIR>/sceregenparams.properties_<PLV> <LOGDIR>/sceregenparams.properties`.

For example, `PLV=BIA_11` in `<LOGDIR>/sceregenparams.properties_BIA_11`. File `<LOGDIR>/sceregenparams.properties` must contain the level 2 folders to regenerate for that adaptor. Examples of level 2 folders are `SILOS`, `PLP` for `BIA_11`, `SDE_JDEE1_91_ADAPTOR` for `JDE_9_1`, `SDE_PSFT_92_ADAPTOR` for `PSFT_9_2`.

18. Run the import in `MODE=SCEREGEN` substituting the appropriate PLV code and source technology.

For example:

BIA_11:

```
/scratch/jdk/jdk_1.8.0_201/bin/java -client -Dfile.encoding=UTF-8
-classpath
```

```

"/scratch/product/Oracle/Middelware/Oracle_Home/odi/sdk/lib/*:/
scratch/product/Oracle/Middelware/Oracle_Home/oracle_common/jlib/*:/
scratch/product/Oracle/Middelware/Oracle_Home/oracle_common/modules/
oracle.idm/*:/scratch/product/Oracle/Middelware/Oracle_Home/oracle_common/
modules/oracle.jps/*:/scratch/product/Oracle/Middelware/Oracle_Home/
oracle_common/modules/oracle.jdbc/*:/scratch/product/Oracle/Middelware/
Oracle_Home/oracle_common/modules/oracle.nlsrtl/*:/scratch/product/Oracle/
Middelware/Oracle_Home/oracle_common/modules/oracle.ucp.jar:/scratch/
product/Oracle/Middelware/Oracle_Home/bi/biapps/lib/*:/scratch/product/
Oracle/Middelware/Oracle_Home/bi/biapps/biaodiutil/lib/bia-odi-
util.jar"
-Dcommon.components.home=/scratch/product/Oracle/Middelware/
Oracle_Home/oracle_common/
oracle.apps.biaodiutil.Import MODE=SCEREGEN
FILEDIR=/scratch/product/Oracle/Middelware/Oracle_Home/bi/biapps/
admin/provisioning/odi/
PLV=BIA_11
JPSCONFIGFILE=/scratch/product/Oracle/Middelware/Oracle_Home/odi/
studio/bin/jps-config-jse.xml
LOGDIR=/scratch/product/Oracle/Middelware/Oracle_Home/bi/biapps/
admin/provisioning/odi/logs/

```

PSFT_9_2 for MSSQL:

```

/scratch/jdk/jdk_1.8.0_201/bin/java -client
-Dfile.encoding=UTF-8 -classpath
"/scratch/product/Oracle/Middelware/Oracle_Home/odi/sdk/lib/*:/
scratch/product/Oracle/Middelware/Oracle_Home/oracle_common/jlib/*:/
scratch/product/Oracle/Middelware/Oracle_Home/oracle_common/modules/
oracle.idm/*:/scratch/product/Oracle/Middelware/Oracle_Home/oracle_common/
modules/oracle.jps/*:/scratch/product/Oracle/Middelware/Oracle_Home/
oracle_common/modules/oracle.jdbc/*:/scratch/product/Oracle/Middelware/
Oracle_Home/oracle_common/modules/oracle.nlsrtl/*:/scratch/product/Oracle/
Middelware/Oracle_Home/oracle_common/modules/oracle.ucp.jar:/scratch/
product/Oracle/Middelware/Oracle_Home/bi/biapps/lib/*:/scratch/product/
Oracle/Middelware/Oracle_Home/bi/biapps/biaodiutil/lib/bia-odi-
util.jar"
-Dcommon.components.home=/scratch/product/Oracle/Middelware/
Oracle_Home/oracle_common/
oracle.apps.biaodiutil.Import MODE=SCEREGEN
FILEDIR=/scratch/product/Oracle/Middelware/Oracle_Home/bi/biapps/
admin/provisioning/odi/
PLV=JDE_9_1 SRCTECH=MICROSOFT_SQL_SERVER
JPSCONFIGFILE=/scratch/product/Oracle/Middelware/Oracle_Home/odi/
studio/bin/jps-config-jse.xml
LOGDIR=/scratch/product/Oracle/Middelware/Oracle_Home/bi/biapps/
admin/provisioning/odi/logs/

```

19. Enter the ODI Repository connection details when prompted.
20. Monitor the process by viewing the command line window or the log files being written to the logs directory.

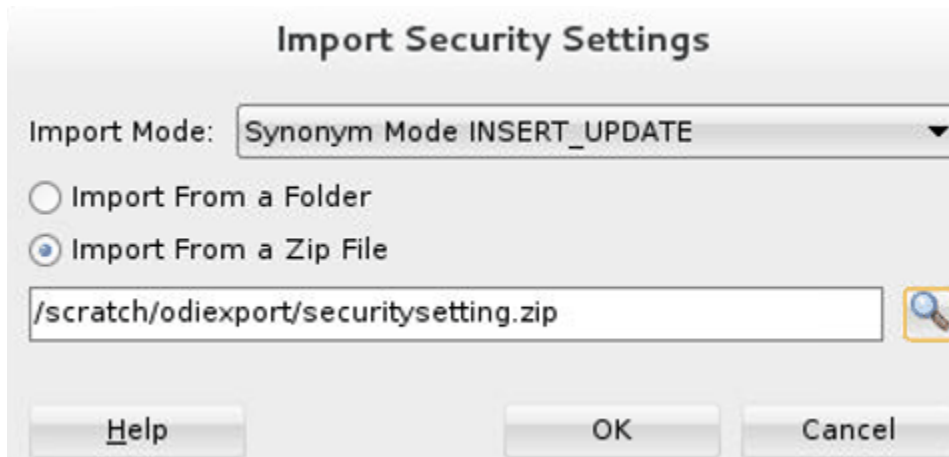
This process can be lengthy and can last from 1 to 10 hours, depending on the number of adaptors for which you're importing the metadata content.

Importing Security Settings in to the ODI Repository

You can import security settings in to the ODI repository.

In this procedure you import the security settings that you exported in [Exporting Security Settings from Original ODI Repository](#).

1. Log in to ODI Studio.
2. Navigate to **Security**, and select **Import Security Settings**.
3. In the Import Security Settings dialog, select **Import mode as Synonym Mode INSERT_UPDATE**, select **Import From a Zip file**, browse for the export file, and then click **OK**.



4. Confirm that there are no issues in the Import report and close it.

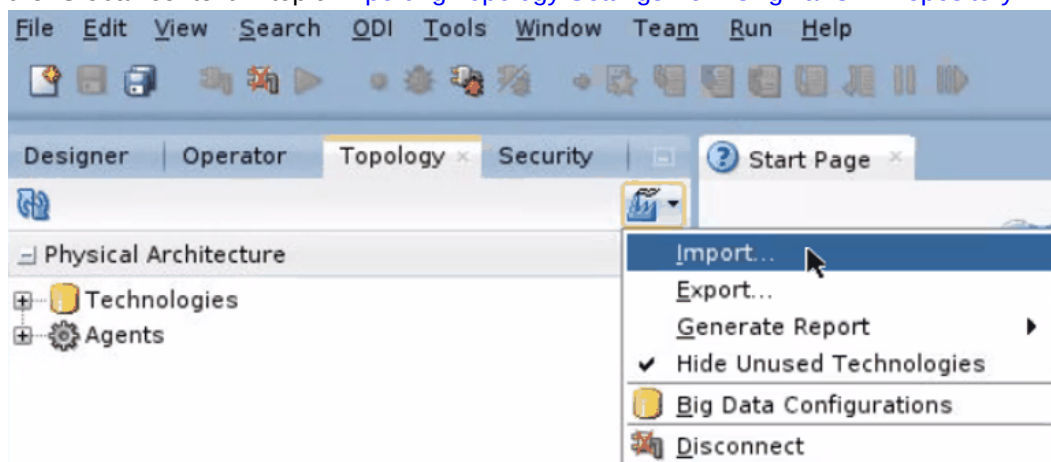
You can now log into the ODI Repository with the security settings configured when the ODI Repository was created using the Oracle Business Analytics Suite RCU.

Importing Connection Details that Configure the Data Servers

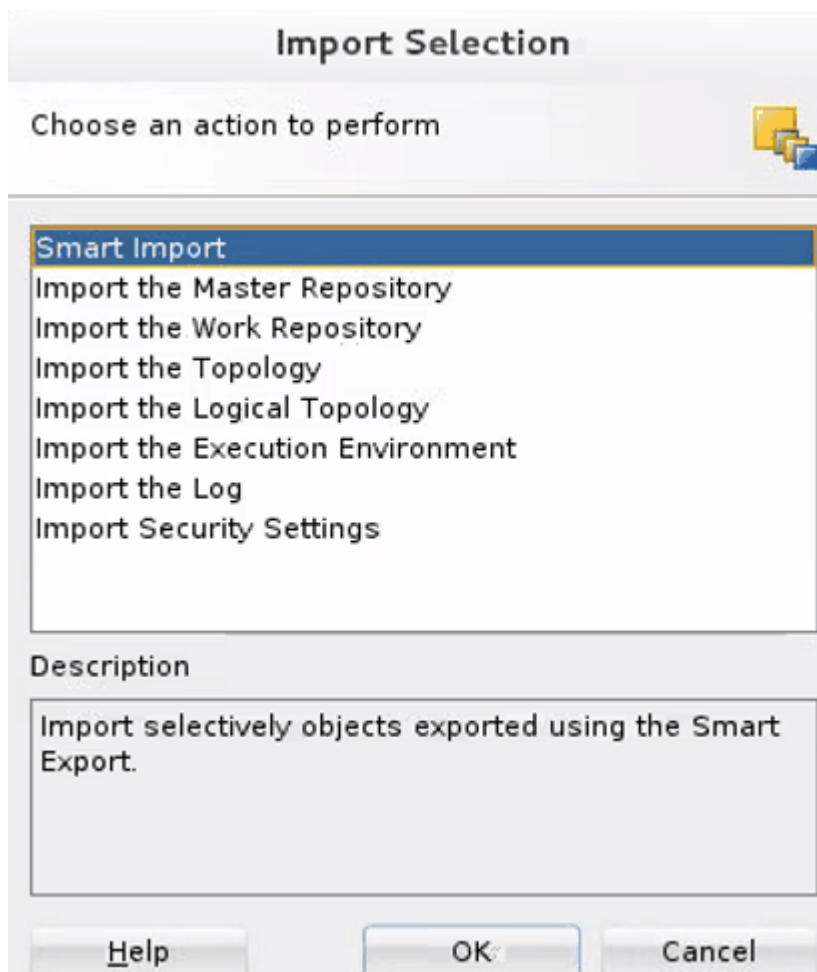
Use this procedure to perform post-import configurations.

1. Launch the ODI Studio client and connect to the ODI repository.
2. Navigate to the Topology tab and complete these steps:
 - a. Expand **Physical Architecture**, click **Technologies**, click **Oracle**, and then click the **BIAPPS_DW** data server.
 - b. Double-click and open your physical schema such as BIAPPS_DW.BIA_DW and set the physical and work schema to match your warehouse schema details.
 - c. Click **Save** and close.
3. Navigate to the Topology tab. From the Connect Navigator (Topology icon dropdown on the top right side of the navigator pane), select **Import**.

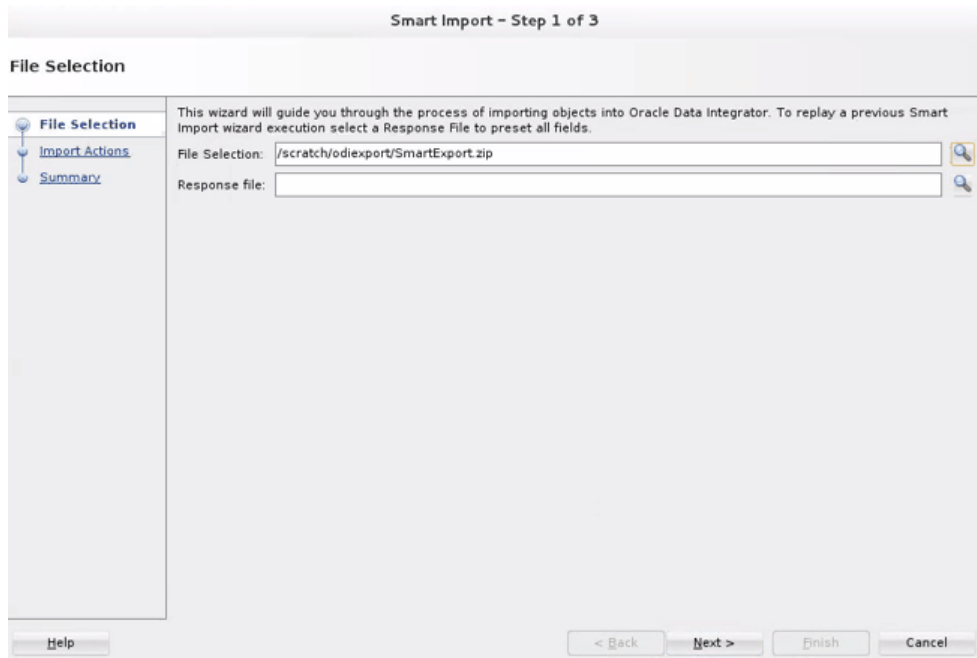
As part of the procedures described below, you import the file created by exporting the 'Global' context in topic [Exporting Topology Settings from Original ODI Repository](#).



4. In the Import Selection dialog, select **Smart Import**.



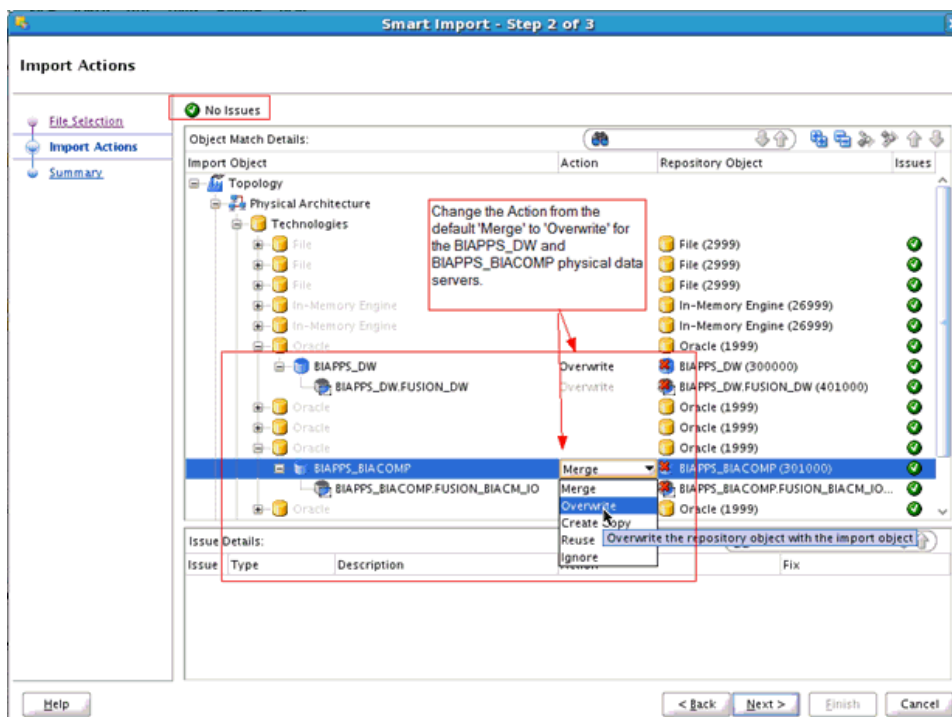
5. In **File Selection**, browse and select the export zip file created in topic [Exporting Topology Settings from Original ODI Repository](#) and click **Next**.



6. In Enter Export Key, provide the export key.



7. The default behavior of Smart Import is to Merge details in the target repository. Ensure that no issues are reported – if issues are reported, resolve them to ensure that the existing details are replaced by the details being imported. Select the **BIAPPS_DW** and **BIAPPS_BIACOMP** Data Servers, select the **Overwrite** option, and continue.



8. Verify the imported details.

Verify that the following logical schemas have the correct data source Num ID value set in the **DATASOURCE_NUM_ID** flex field:

- DW_BIAPPS11G

Name	Default	Value
PRODUCT_LINE_VERSION_KEY	<input type="checkbox"/>	BIA_11
DATA_SERVER_TYPE	<input type="checkbox"/>	BIA_11
DATASOURCE_NUM_ID	<input type="checkbox"/>	999
OBI Schema Type	<input type="checkbox"/>	DW

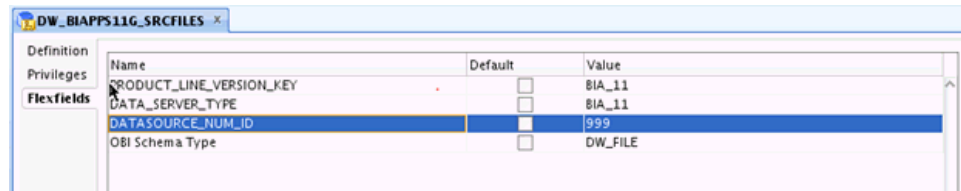
- DW_BIAPPS11G_LOG_FILES

Name	Default	Value
PRODUCT_LINE_VERSION_KEY	<input type="checkbox"/>	BIA_11
DATA_SERVER_TYPE	<input type="checkbox"/>	BIA_11
DATASOURCE_NUM_ID	<input type="checkbox"/>	999
OBI Schema Type	<input type="checkbox"/>	DW_LOG_FILE

- DW_BIAPPS11G_SHARED_FILES

Name	Default	Value
PRODUCT_LINE_VERSION_KEY	<input type="checkbox"/>	BIA_11
DATA_SERVER_TYPE	<input type="checkbox"/>	BIA_11
DATASOURCE_NUM_ID	<input type="checkbox"/>	999
OBI Schema Type	<input type="checkbox"/>	DW_SHARED_FILE

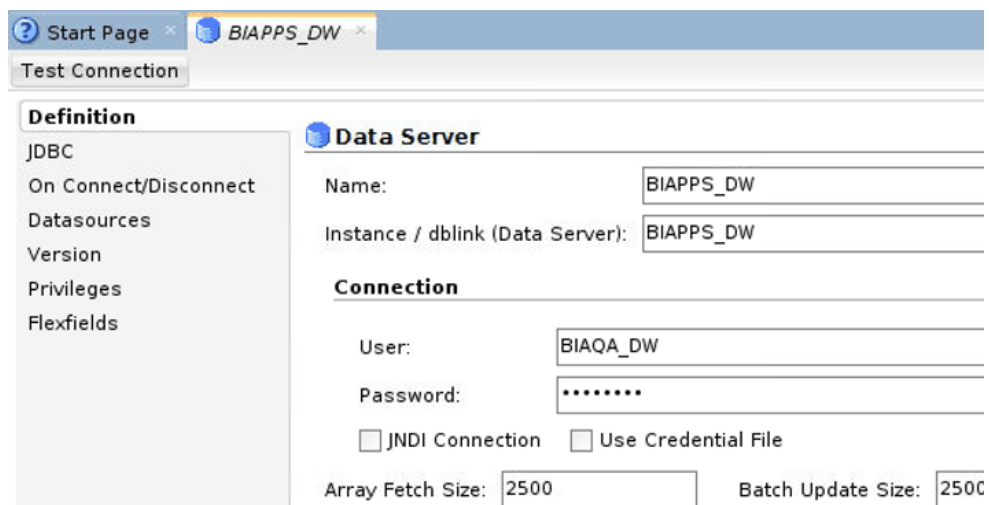
- DW_BIAPPS11G_SRCFILES



Definition	Name	Default	Value
Privileges	PRODUCT_LINE_VERSION_KEY	<input type="checkbox"/>	BIA_11
Flexfields	DATA_SERVER_TYPE	<input type="checkbox"/>	BIA_11
	DATA_SOURCE_NUM_ID	<input checked="" type="checkbox"/>	999
	OBI Schema Type	<input type="checkbox"/>	DW_FILE

If you don't see the value 999 for all these logical schemas, then set the values to 999.

- In ODI Studio, click **Topology** and select **Physical Architecture**. Open the BIAPPS_DW physical server.
- Verify that the **User** and **Password** are correctly populated on the Definition tab.



Definition

JDBC

On Connect/Disconnect

Datasources

Version

Privileges

Flexfields

Data Server

Name: BIAPPS_DW

Instance / dblink (Data Server): BIAPPS_DW

Connection

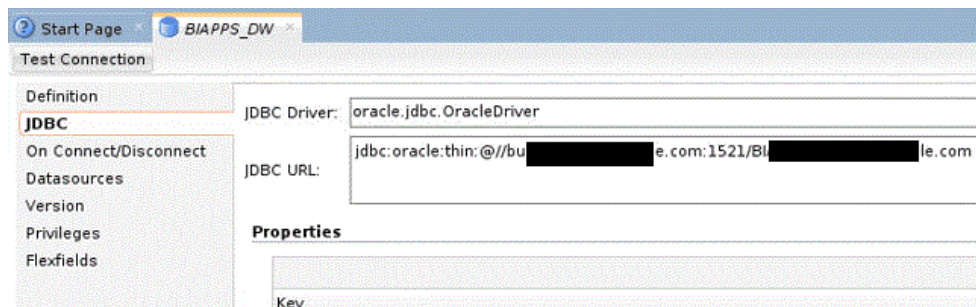
User: BIAQA_DW

Password:

JNDI Connection Use Credential File

Array Fetch Size: 2500 Batch Update Size: 2500

- On the JDBC tab, verify that the **JDBC URL** is correctly set. If you used the default **Merge** action during the Smart Import, then the **User** and **Password** are updated but the **JDBC URL** remains unchanged.



JDBC

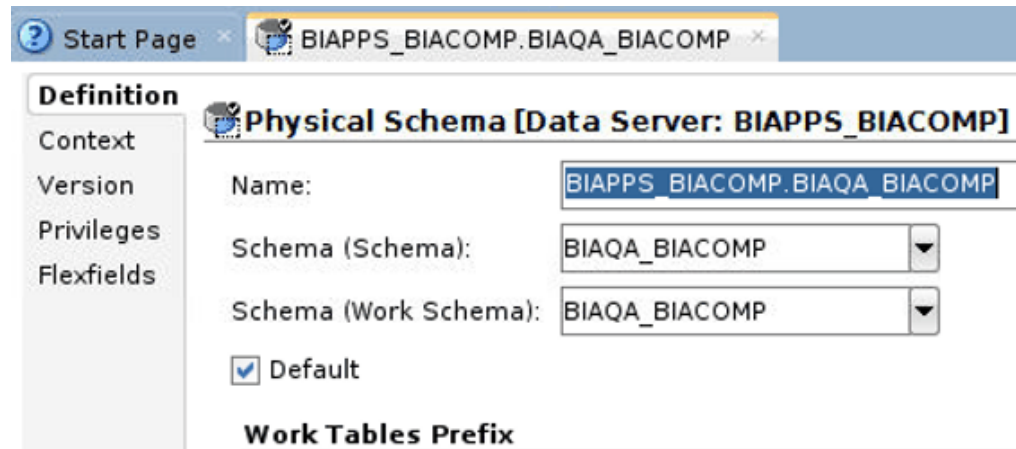
JDBC Driver: oracle.jdbc.OracleDriver

JDBC URL: jdbc:oracle:thin:@//bu[redacted].e.com:1521/BI[redacted].le.com

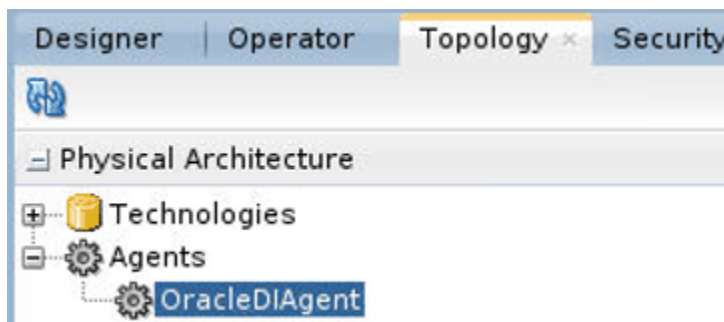
Properties

Key

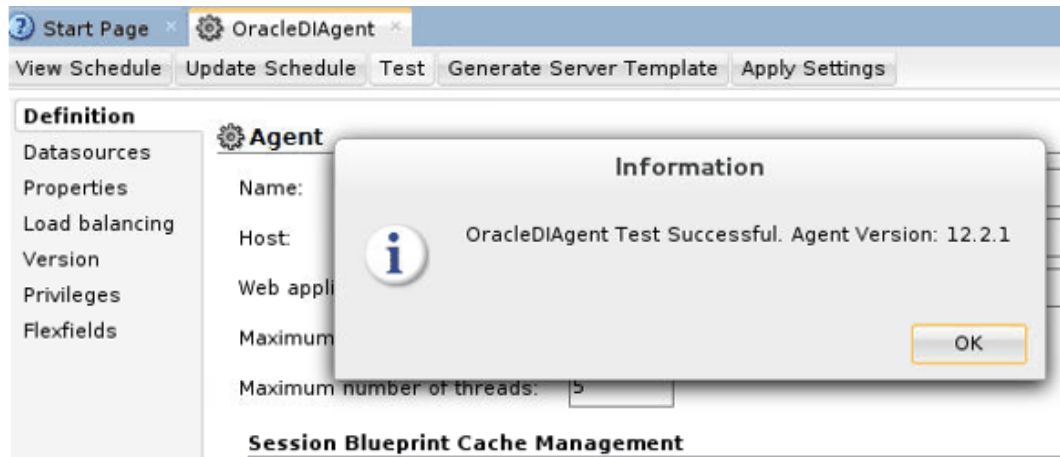
- Repeat steps 1 through 4 for the BIAPPS_BIACOMP physical server.
- Verify that the Physical schema for BIACOMP and DW are set as defaults. If they aren't set as defaults, then the load plans fail.
 - In ODI Studio, click **Topology** and select **Physical Architecture**.
 - Open the physical schema under the BIAPPS_BIACOMP physical server.



- c. Select the **Default** check box if it isn't selected on the Definition tab.
 - d. Save your changes.
14. Repeat steps 2 through 4 for the physical schema under the `BIAPPS_DW` physical server.
15. In ODI Studio, click Topology, select Physical Architecture, and then expand Agents.



16. Right-click on the configured agent and then click **Open**.



A

Generating DDL and Assigning Tablespaces to Tables and Indexes

These topics describe how to generate DDL to deploy Oracle Business Analytics Warehouse tables and how to assign tablespaces to tables and indexes.

Topics:

- [Overview](#)
- [Generating the Business Analytics Warehouse DDL](#)
- [Patching Oracle BI Applications](#)
- [Assigning Tablespaces](#)

Overview of Oracle Business Analytics Warehouse Tables

The Oracle Business Analytics Warehouse tables are automatically deployed during the installation process when the Business Analytics Applications Suite Repository Creation Utility (RCU) executes a shipped DDL script.

The RCU doesn't prompt for which tablespace to assign to the individual tables and related indexes nor does it provide a mechanism for you to alter the shipped DDL. To introduce changes to the Oracle Business Analytics Warehouse data model, you use Oracle Data Integrator (ODI) to generate a new DDL script.

You may want to regenerate the Oracle Business Analytics Warehouse DDL for these reasons:

- You may want to modify the Oracle Business Analytics Warehouse, either by modifying existing tables or creating new ones.
- You may want to have separate development environments. In this case you must use ODI to generate the DDL to be executed in these environments to deploy the Oracle Business Analytics Warehouse tables rather than run the RCU for each environment.
- The default tablespace assigned to the Oracle Business Analytics Warehouse database user is used for all tables and indexes when those objects are created by the RCU. You may want to assign different tablespaces to the DW tables other than the default tablespace associated with the database user.
- If a patch or upgrade is applied that alters an ODI data store or introduces a new ODI data store, the corresponding database table must be altered or created.

You can make changes directly in the database but these changes must be synchronized with the ODI Repository. Run the Reverse Knowledge Module (RKM) of Oracle BI Applications to bring in the changes into ODI. If this RKM doesn't run, ODI won't be aware of these changes, which can lead to problems when populating the affected warehouse table.

Generating the Oracle Business Analytics Warehouse DDL

Use this procedure to generate the Oracle Business Analytics Warehouse DDL.

Keep these items in mind when introducing changes in ODI:

- Full mode generates a script that creates all tables and sequences, regardless of what may already be deployed in the database. Incremental mode compares the ODI and database definition of the tables and columns. Neither mode creates, modifies, or drops indexes.
- Add or modify columns:
 - Add column: ODI allows duplicate column names but you get an error when deploying the DDL to the database.
 - Modify column:
 - Column renaming in ODI is treated as a drop of the existing column and a new column is added in the database. This means any existing data in that column is lost. The procedure works on column name. When generating the DDL, the procedure compares the database and ODI definitions of the columns. If a column has been renamed in ODI, that column won't be found in the database, so it's added.
 - Changes to datatype, scale, precision, and nullability are supported as ALTER statements
 - You should avoid altering preconfigured columns. They should introduce their own custom column if they need a change in functionality.
 - Existing columns can be changed to a larger size but shouldn't be reduced to a smaller size. Similarly, column data types can be changed to one that already accommodates existing data (CHAR to VARCHAR, DATE to DATETIME) but shouldn't be changed to less accommodating datatype (VARCHAR to NUMBER). ODI doesn't prevent users from making this kind of change, but when deploying the DDL, users will likely encounter issues.
 - Delete column: You shouldn't delete preconfigured columns. You should only delete custom columns. If a column is deleted from the ODI Repository, it is dropped from the table using an ALTER statement. Identify the columns that exist in the database but don't exist in ODI. (Thus, renaming the column in ODI results in a drop and add statements being generated).
- Add or modify indexes:
 - Add index: ODI allows duplicate index names and duplicate index definitions, but you get an error when deploying the DDL on the database.
 - Modify index:
 - Indexes are only dropped and created during the execution of a load plan. Indexes aren't synchronized by this procedure.
 - Changes to preconfigured indexes is fully supported. Changes to uniqueness and active/inactive are supported. The primary use case is to make a preconfigured index active or inactive
 - Changes to preconfigured index names should be avoided. If a load plan was previously executed and created the index, the index continues to exist

with the old name in the database. Because ODI isn't aware of this index, problems may occur during the execution of the load plan.

- Delete index:
 - You should only delete custom indexes and not preconfigured indexes.
 - If you don't want a preconfigured index, inactivate it.
- Add or drop a table: You can add a new table or delete an existing table in ODI.
 - If a table is added in ODI and doesn't exist in the database, it is added using a CREATE TABLE statement.
 - If a table is deleted from ODI and still exists in the database, a DROP TABLE statement is generated.
 - You shouldn't delete preconfigured tables.
- Add or drop a sequence:

You should add a sequence in ODI for new tables that includes the ROW_WID column and that follows the format <TABLE_NAME>_SEQ. You should add a sequence in ODI for new tables that includes the SCD1_WID column and that follows the <TABLE_NAME>_S1W naming convention.

- If a sequence is added in ODI and doesn't exist in the database, it is added using a CREATE SEQUENCE statement.
 - Sequences deleted from ODI aren't dropped from the database.
1. Launch ODI Studio, and display the Designer navigator.
 2. In the Projects editor, expand these folders: Components, DW, Oracle, and Generate DW DDL.
 3. Execute the GENERATE_DW_DDL scenario:

Specify this information:

Option	Description
CREATE_SCRIPT_FILE	If set to True, this creates a .sql file with the DDL script. This is useful to see what the changes are. If no value is provided for the SCRIPT_LOCATION option, the procedure creates a file in the oracledi directory named biappsddl.sql.
REFRESH_MODE	FULL or INCREMENTAL FULL generates the DDL for the entire data warehouse using CREATE statements. The Table Mask option is ignored. INCREMENTAL compares the ODI data stores with the tables in the database. This option should be used with the Table Mask option because this process can take a very long time comparing the entire data warehouse.
CHARCLAUSE	Provided for Unicode support. If set to True, the CHAR clause is included in the DDL.
RUN_DDL	If set to True, the DDL runs against your warehouse.
SCRIPT_LOCATION	If you're creating a script, this field indicates the path where the script is created.
TABLE_MASK	Default % compares all tables. If you want to create DDL for a subset of tables only, or just a single table, you can change this value. For instance, to create a diff script for dimension tables, you'd set this value to W_%_D.

Oracle BI Applications Patches

Patching Oracle BI Applications may require changes to database objects, such as modifying existing tables or introducing new tables. Oracle provides a patch file that you use to merge changes with the existing ODI Repository definition.

These changes must then be deployed to the database. New tables, columns and indexes that you introduced are retained. Any changes you made to preconfigured tables, columns, and indexes must go through a conflict resolution process.

Assigning Tablespaces

Using the Designer navigator in Oracle Studio, you can assign default tablespaces for the data warehouse tables, such as the W_%_D and W_%_F tables, their associated indexes, and the staging tables, such as the W_%_DS and W_%_FS tables. Indexes aren't created by this utility but are created during the ETL process and reflects the tablespace assigned in ODI.

To assign tablespaces to data warehouse tables:

1. Launch ODI Studio, and display the Designer navigator.
2. In the Models editor, expand the Oracle BI Applications model.
3. Edit the Oracle BI Applications model and assign the OBI Tablespace flexfields with the tablespace values to be used for these objects.

All data warehouse tables and indexes use the tablespaces defined here.

4. (Optional) Individual tables can be overridden to use a different tablespace from those assigned at the model level.

Edit the particular data store and assign the OBI Tablespace flexfield with the tablespace to be used for this table.

5. (Optional) To assign the tablespace for individual indexes, edit the constraint and assign the OBI Tablespace flexfield value to match the tablespace name to be used.

B

Integrating Dashboards and Siebel Applications

Perform these additional configuration steps to run Oracle Business Intelligence Applications (Oracle BI Applications) dashboards with a Siebel application.

After you've configured Oracle Business Intelligence Applications on Oracle Analytics Server and are able to access the dashboards, you must update the Siebel application to view the Analytics dashboards within the Siebel application.

Review and perform these steps to integrate Oracle Business Intelligence Applications and Siebel application:

- [Overview of Embedding Oracle Analytics in Siebel Application](#)
- [Configure Oracle HTTP Server](#)
- [Update the Siebel Application to Find Oracle Analytics Server Through HTTP Server](#)
- [About Configuring Oracle BI Action Links](#)
- [Accessing Optional Analytics Applications](#)
- [Customizing Oracle BI Application Home Page and Dashboards](#)
- [Viewing Information About Preconfigured Dashboards](#)

Overview of Embedding Oracle Analytics in Siebel Application

You can embed your analytics content into applications and web pages.

If, for example, you plan to run Siebel Web Extension (SWE) and Oracle Analytics Server on different web servers, you must use some kind of networking or load balancing mechanism to create a single logical domain (or virtual IP address) for the two machines. One way to achieve this is to install a supported HTTP Server such as Oracle HTTP Server. You then configure the HTTP server against both the Siebel Web Extension (SWE) and the entry point to Oracle Analytics. In this way, you configure a reverse proxy that makes it appear to client browsers that both Siebel application and Oracle Analytics are being served from the same machine.

After modifying the HTTP server's proxy settings, you must update the Siebel application administration settings to map the HTTP server to the NQHOST and NQHOSTHOME variables. Finally, you must confirm that each URL contains a properly formatted reference to the /analytics directory.

 **Note:**

The URL to both Siebel application and Oracle Analytics must use the same protocol. In other words, both must use HTTP, or both must use HTTPS.

When displaying Oracle Analytics content in any other portals using iFrames, you might see the following message "Oracle Analytics content cannot be displayed in an iFrame".

To display Oracle Analytics content on a web page using iFrame, set different values to the `IniFrameRenderingMode` element in the `instanceconfig.xml` file. See [Protect Pages in Oracle Analytics Server from Attack](#).

Configure Oracle HTTP Server

When Oracle Analytics Server and Siebel application are running on different web servers, install a supported HTTP server such as Oracle HTTP Server to configure a reverse proxy so that the client browser thinks the Oracle Analytics Server and the Siebel application are on the same machine.

This procedure assumes that you are using Oracle HTTP Server to embed Oracle Analytics in the Siebel application. If you are working in a production environment, Oracle recommends that you add security policies to the HTTP server so that users cannot directly access the underlying business intelligence objects. Note that the HTTP server must run on the same port as the Siebel application server.

1. Open Oracle HTTP Server's `mod_wl_ohs.conf` file.
2. Insert the following proxy request information into the configuration file.

For example:

```
ProxyRequest Off
<Proxy*>
Order deny,allow
Allow from all

</Proxy*>
Location /Oracle Siebel application subdirectory
ProxyPass http://location of Oracle Siebel application server/
Oracle Siebel application subdirectory
ProxyPassReverse http://location of Oracle Siebel application
server/
Oracle Siebel application subdirectory
/Location

Location /Oracle Analytics subdirectory
ProxyPass http://location of Oracle Analytics Server/
Oracle Analytics subdirectory
ProxyPassReverse http://location of Oracle Analytics Server/
Oracle BI EE subdirectory
/Location
```

3. Save and close the configuration file.
4. Test the configuration by using a fully-qualified name to log into Oracle Siebel application through Oracle HTTP Server.

Update the Siebel Application to Find Oracle Analytics Server Through HTTP Server

Use this procedure to change the Siebel application's settings to map the HTTP server to the NQHOST and NQHOSTHOME variables.

1. Open your Siebel application.
2. Select **Navigate**, and then **Site Map**.
3. Click **Administration - Integration**.
4. Click **Host Administration view**.
5. Query for NQHOST in the Virtual Name column:

In the **Name** column, update the name to the host name of the HTTP server that is serving Oracle Analytics Server.

This is the location name that you specified in the Oracle HTTP Server's `mod_wl_ohs.conf` file. For information about this location, see [Configure Oracle HTTP Server](#).

Note that if the HTTP server is running on a port other than 80, you must include the port number. For example, `servername.oracle.com:7003`.

6. Log out of the application and log back in.

About Configuring Action Links

You can create action links so users can navigate to related BI content or perform business tasks.

To configure and use BI action links, see *Configure for Actions with the Action Framework in Administering Oracle Analytics Server*.

Access Optional Analytics Applications

Depending on the options you purchased with your Siebel application, you must perform additional steps in order to access the corresponding Oracle Analytics Server options.

Additional options for Oracle Analytics Server are:

- Oracle Sales Analytics
- Oracle Service Analytics
- Oracle Marketing Analytics
- Oracle Price Analytics
- Oracle Partner Analytics

By default, the dashboards and reports contained in these optional areas are hidden. For example, if you purchased Sales Analytics with your Sales application, you must perform the steps in Assign Access Permissions to access the Sales Analytics catalog objects.

Customize Oracle BI Applications Home Page and Dashboards

You can customize the home page of Oracle BI Applications and dashboards.

1. [Configuring a New Home Page to Use a Symbolic URL.](#)
2. [Determining the Oracle BI Report Path Argument.](#)
3. [Adding Views for Custom Oracle BI Interactive Dashboards.](#)

Configure a New Home Page to Use a Symbolic URL

Oracle BI Applications are integrated with Siebel applications using the symbolic URL infrastructure. This task describes how to use symbolic URLs to link a new report to a Siebel application home page and how to add new Analytics dashboards to the Siebel application. The symbolic URL specifies how the HTTP request to the external application should be constructed and defines any arguments and values to be sent as part of the request.

This information assumes that you've successfully built a symbolic URL to link some external content.

For each Analytical report on a Siebel application home page, there is a symbolic URL record defined that links the home page to the Analytics report. If you haven't already done so, you must create a new home page and set it up to use a symbolic URL.

This figure shows example Inline and iFrame symbolic URL arguments:

Inline

Name	Required Argument	Argument Type	Argument Value	Append as Argument	Substitute in Text	Sequence #▲
Cmd	<input checked="" type="checkbox"/>	Constant	Go	<input checked="" type="checkbox"/>		1
Path	<input checked="" type="checkbox"/>	Constant		<input checked="" type="checkbox"/>		2
Syndicate	<input checked="" type="checkbox"/>	Constant	Siebel	<input checked="" type="checkbox"/>		3
logUser	<input checked="" type="checkbox"/>	Command	UseSiebelLoginId	<input checked="" type="checkbox"/>		4
logPassword	<input checked="" type="checkbox"/>	Command	UseSiebelLoginPassword	<input checked="" type="checkbox"/>		5

iFrame

Name	Required Argument	Argument Type	Argument Value	Append as Argument	Substitute in Text	Sequence #▲
iFrameLoginCmd	<input checked="" type="checkbox"/>	Constant	Logon	<input checked="" type="checkbox"/>		1
Cmd	<input checked="" type="checkbox"/>	Constant	PortalPages	<input checked="" type="checkbox"/>		2
iFrameLoginUser	<input checked="" type="checkbox"/>	Command	UseSiebelLoginId	<input checked="" type="checkbox"/>		3
iFrameLoginPassword	<input checked="" type="checkbox"/>	Command	UseSiebelLoginPassword	<input checked="" type="checkbox"/>		4
PortalPath	<input checked="" type="checkbox"/>	Constant	/shared/Service/_portal/Activities	<input checked="" type="checkbox"/>		5
PostRequest	<input checked="" type="checkbox"/>	Command	PostRequest	<input checked="" type="checkbox"/>		6
iFrameLoginSyndicate	<input checked="" type="checkbox"/>	Constant	Siebel	<input checked="" type="checkbox"/>		7

To configure a new Home Page to use a symbolic URL:

1. Make sure the symbolic URL has been set up.
2. Launch the Siebel application and navigate to the Administration - Integration, then Symbolic URL Administration view.
3. Query for the symbolic URL that has been set up for Analytics.

Update the name to the host name of the HTTP server that is serving the Oracle Analytics service.

The name of this symbolic URL should be exactly the same as the calculated value of the field that was added to the Business Component. For example, you might have a symbolic URL named HomePageAnalytics.

4. In the URL field, enter the Web URL. For example:

`http://NQHOST/analytics`

For the Host Name, choose the HTTP server that is serving the Oracle Analytics service. This table shows the other parameters for reports.

Parameter	Value
NQHOST	Host name of the HTTP sever that is serving the Oracle Analytics service.
Fixup Name	Inside Applet
SSO Disposition	Inline
Oracle Analytics Presentation Services application name	Select from the drop-down list

5. Create the appropriate symbolic URL Arguments.

These arguments depend upon the particular Analytics report that you're trying to display. The Argument Values should be the same for any Analytics report you work with, except for the Path Argument Value. To determine the path to the Analytics report, see [Determining the Oracle BI Report Path Argument](#).

Determine the Oracle Analytics Report Path Argument

The Path argument provides the symbolic URL with the path to a report in Oracle Analytics Server. (For example, /shared/Sales/Analytic Library/Pipeline/Top Current Quarter Opportunities.)

1. Log in to Oracle Analytics Server as an Administrator.
2. In Oracle Analytics Server, navigate to **Catalog**, and then navigate to your report.
3. Add the report path to the symbolic URL argument.

This table shows the symbolic URL path arguments for reports.

Name	Type	Path Argument Value	Append	Sequence #
Cmd	Constant	GO	Y	1
Path	Constant	/shared/Sales/Analytic Library/Pipeline/Top Current Quarter Opportunities	Y	2

Name	Type	Path Argument Value	Append	Sequence #
Syndicate	Constant	Siebel	Y	3
nQUser	Command	UseSiebelLoginId	Y	4
nQPassword	Command	UseSiebelLoginPassword	Y	5
PostRequest	Command	PostRequest	Y	6

Add Views for Custom Dashboards

You can add views for custom dashboards.

1. Use Siebel Tools and set up a new view.
2. In the Siebel application, define a symbolic URL for that view.
3. Define a symbolic URL.
 - a. Navigate to **Site Map**, then **Administration - Integration**, then **Symbolic URL Administration**.
 - b. In the Symbolic URL Administration view, add a new record.
4. Define the symbolic URL arguments.
 - a. Navigate to **Symbolic URL Administration**.
 - b. In the Symbolic URL Administration list, select the symbolic URL you want to configure.

View Preconfigured Dashboards

If your organization has prebuilt applications installed, you can use Catalog Manager to locate and view the preconfigured dashboards.

You might have to expose the dashboards and requests.

Exposing Dashboards and Requests

Depending on the Oracle Analytics options your organization purchased, you might need to expose these options before the associated dashboards and requests can be viewed in Oracle Analytics Server and in Catalog Manager.

This applies to sites that have these prebuilt applications:

- Sales Analytics
- Service Analytics
- Contact Center Analytics
- Marketing Analytics
- Partner Analytics

Locating Dashboards, Page Names, and Reports

In Catalog Manager, the Catalog distributed with prebuilt applications has this structure: Catalog followed by shared folder, prebuilt application name, `_portal` folder, dashboard name, and page name.

The path to locate reports is Catalog followed by shared folder, prebuilt application name, folder name, and report name.

C

Configuring Oracle Analytics Server Domain

This topic describes how you can configure the Oracle Analytics Server domain.

Topics:

- [Create BI Schema on Oracle ATP Database](#)
- [Configure Oracle Analytics Server Domain Using Configuration Assistant](#)
- [Configure Oracle Analytics Server Domain on Oracle ATP](#)

Create BI Schema on Oracle ATP Database

You can create the BI schema on Oracle ATP either by using the RCU in graphical user interface (GUI) or in silent mode.

If you use the RCU in GUI, at several intervals the RCU might display insufficient privileges message. Ignore this message and continue with the next step in the wizard. During Oracle Analytics Server RCU creation, select DATA for Default Tablespace and TEMP for Temp Tablespace. Autonomous transaction processing database provide various services consumer groups. Oracle recommends you to always use "TP" ATP service name while running RCU and configuring Oracle BI Applications.

Perform the following steps to run the RCU in silent mode.

1. Run the RCU in silent mode with the following options in the response file, and input the full TNS connection string to create the schema.

```
rcu -silent -responseFile <path-to-the-response-file>
```

- **connectString for a connection with wallet:**

```
description= (retry_count=20) (retry_delay=3) (address=(protocol=tcps)
(port=1522) (host=adb.uk -london-1.oraclecloud.com))
(connect_data=(service_name=vljgk4wx2kcrt do_bi_tp.a
db.oraclecloud.com)) (security=(ssl_server_cert_dn="CN=adwc.eucom-
central-1.oraclecloud.com, OU=Oracle BMCS FRANKFURT, O=Oracle
Corporation, L=Redwood City, ST=California, C=US")))
```

- **connectString for a connection without wallet:**

```
(description= (retry_count=20) (retry_delay=3) (address=(protocol=tcps)
(port=1522) (host=adb.uk -london-1.oraclecloud.com))
(connect_data=(service_name=vljgk4wx2kcrt do_bi_tp.a
db.oraclecloud.com)) (security=(ssl_server_dn_match=yes)))
```

- `operation=createRepository`

- `databaseType=ORACLE`

- dbUser=<DB User>
- dbRole=Normal
- schemaPrefix=<Schema Prefix>
- componentList=STB,OPSS,BIPLATFORM,IAU,IAU_APPEND,IAU_VIEWER,MDS,WLS
- selectDependentsForComponents=true
- useSamePasswordForAllSchemaUsers=true
- STB.tablespace=DATA
- STB.tempTablespace=TEMP
- OPSS.tablespace=DATA
- OPSS.tempTablespace=TEMP
- BIPLATFORM.tablespace=DATA
- BIPLATFORM.tempTablespace=TEMP
- IAU.tablespace=DATA
- IAU.tempTablespace=TEMP
- IAU_APPEND.tablespace=DATA
- IAU_APPEND.tempTablespace=TEMP
- IAU_VIEWER.tablespace=DATA
- IAU_VIEWER.tempTablespace=TEMP
- MDS.tablespace=DATA
- MDS.tempTablespace=TEMP
- WLS.tablespace=DATA
- WLS.tempTablespace=TEMP

2. After successful creation of schema, continue with the Install Oracle Data Integrator section. See [Install Oracle Data Integrator](#).

Configure Oracle Analytics Server Domain Using Configuration Assistant

Use the Configuration Assistant (.sh or .cmd file) to configure the Oracle Analytics Server domain.

If you are extending the WebLogic domain with Oracle Analytics Server by using the Configuration Assistant, make sure that the Administration Server for the domain is named AdminServer. Other names for the Administration Server aren't supported.

To configure Oracle Analytics Server on Oracle ATP, see [Configure Oracle Analytics Server Domain on Oracle ATP](#).

For the list of databases supported by Oracle Analytics Server, see Certification - Supported Data Sources in *Administering Oracle Analytics Server*.

1. Change to the bi/bin directory.
 - (Linux) `ORACLE_HOME/bi/bin`
 - (Windows) `ORACLE_HOME\bi\bin`
2. Start the Configuration Assistant.
 - (Linux) `./config.sh`
 - (Windows) `config.cmd`
3. Select the components to install and click **Next**.

To ensure that a consistent set of suites are deployed, the Configuration Assistant automatically adjusts your selection.

- Oracle Analytics Server: Includes components such as data visualization, analyses, dashboards and agents, and installs Oracle Analytics Server without Publisher.
- Oracle Analytics Publisher: Includes Publisher for pixel-perfect reports. You can use this option to install standalone Publisher.

Select both Oracle Analytics Server and Oracle Analytics Publisher components to install Oracle Analytics Server with Publisher.

4. On the Prerequisite Checks screen, after the prerequisite checks conclude with no errors, click **Next**.

If any of the prerequisite checks fail, then a short error message appears at the bottom of the screen. Fix the error and click **Rerun** to try again. To ignore the error or warning messages and continue with the installation, click **Skip**, although this approach is not recommended.

Note:

The configuration might not function normally if you continue without resolving the issue that caused an error or warning message during the prerequisite checks.

5. On the Define New Domain screen, specify the following:
 - Enter the location where you want to set up the domain directory. The new domain is stored in a sub-directory with the same name as the domain.
 - Enter a name for the domain. You can use alphanumeric characters, underscores or hyphens.
 - Enter a username for the default system administrator. The username must not be blank and shouldn't exceed 30 characters.

 **Note:**

This user is created in the embedded LDAP and is granted WebLogic Administrator permissions.

- Enter a password for the system administrator. It must begin with a letter, include at least one number, and can contain alphanumeric, underscore, dollar sign, and pound sign characters.
 - Reenter the password to confirm. **Domain Home** is a read-only field derived from 'Domains Directory' and 'Domain Name'.
6. Click **Next** to proceed to the Database Schema screen. On this screen, click the appropriate option either to create a new schema or use an existing schema.
 - If you select to create a new schema, specify the following:

 **Note:**

In this case, the Configuration Assistant creates a schema for you.

- a. Enter a unique schema prefix.
 - b. Enter a password for your schema.
 - c. Re-enter the password to confirm.
 - d. Select the type of database to use from the list.
 - e. Enter the username of the account that has privileges to create new schemas.
 - f. Enter the password for the username.
 - g. Enter the connect string based on the type of database you are using.
- If you select to use an existing schema, specify the following:

 **Note:**

In this case, you must create STB, BIPLATFORM, IAU, MDS, OPSS, and WLS schemas using the RCU.

- a. Select the type of database to use from the list.
- b. Enter the connect string based on the type of database you are using.

- c. Enter a unique schema prefix.
 - d. Enter a password for your schema.
7. On the Port Management screen, specify the port range and click **Next**.

 **Note:**

The default allocated port range is from 9500 to 9999, both inclusive. You can keep the default values or specify different values within this range.

8. On the Initial Content screen, select one of the following options, and click **Next**:
 - Existing export archive (.jar file) from a previous installation.

 **Note:**

This option is applicable if you are migrating from Oracle BI 11g to Oracle Analytics Server. See *Migrating and Upgrading Oracle Analytics Server*.

- Clean Installation (default).
9. On the Summary screen, verify the values that you specified on each screen.
 - Click **Save** to generate a response file used for silent installation (optional).
 - Click **Configure**.
The configuration process starts and the Configuration Progress screen is displayed.
 10. After the configuration concludes without any errors, click **Next** to go to the Configuration Complete screen.
 11. On the Configuration Complete screen, review the configuration summary.
 - Click **Save** to save the information displayed on this screen in a file.
 - Click **Finish** to close the Configuration Assistant.
The Oracle Analytics Server Application opens in the browser. Use the login credentials that you specified while configuring to sign in to the Home page of Oracle Analytics Server.

You have configured the standard Oracle Analytics Server topology. The Configuration Assistant automatically starts the newly configured Oracle Analytics Server instance after successful completion. However, if you want to restart a domain that has been shut down manually, start servers and processes. After successful configuration of Oracle Analytics Server, continue with the Edit Your Input File section. See [Edit Your Input File](#).

Configure Oracle Analytics Server Domain on Oracle ATP

Run the Configuration Assistant in silent mode to configure the Oracle Analytics Server standard topology on Oracle ATP.

 **Note:**

The Configuration Assistant in silent mode supports only the TNS_ADMIN type connection string. Also, the TNS_ADMIN type connection string is supported only for SCHEMA_TYPE_EXISTING. The Oracle Autonomous Transaction Processing (ATP) database provides various services consumer groups. Oracle recommends you to always use "TP" ATP service name while running RCU and configuring Oracle BI Applications.

1. Extract the downloaded ATP wallet file to a new folder (for example, /scratch/wallet).
2. For silent mode, prepare a response file.

Add the following parameters in the response file. These are examples. Make the appropriate changes as per your environment.

```
[ENGINE]
#DO NOT CHANGE THIS.
Response File Version=1.0.0.0.0
[GENERIC]
#If false, omit the startup action at the end of the config
assistant.
PERFORM_BI_STARTUP=true
SERVICE_INSTANCE_LIMIT=2
#If empty string will default to BI logout page
LOGOUT_URL=
DOMAIN_NAME=bidomain
DOMAINS_DIR=/scratch/Oracle/installation/Middleware/user_projects/
domains
CONFIGURE_BIEE=true
CONFIGURE_BIP=true
SERVICE_INSTANCE_MODE=SINGLE
DOMAIN_WORK=LOCAL
INSTALL_EXALYTICS_IN_MEMORY_SOFTWARE=false
ADMIN_USER_NAME=WebLogic admin user name>
ADMIN_PASSWORD=<WebLogic admin user password>
ADMIN_CONFIRM_PASSWORD=<WebLogic admin user password>
SCHEMA_TYPE=SCHEMA_TYPE_EXISTING
EXISTING_DATABASE_TYPE=ORACLE_TNS_ALIAS
DATABASE_TYPE=ORACLE
FAN_CONNECT_STRING=<can be left blank>
#In Windows, when you configure the Oracle Analytics Server domain,
provide the
# connect string in the format below. Use "\\" as the path
separator.
#Example: #CONNECT_STRING=biaoas2023_tp?TNS_ADMIN=C:\\biapps\
```

```

\atp_wallet
CONNECT_STRING=service_name?TNS_ADMIN=<ATP wallet extract folder>
EXISTING_DB_PREFIX=<Schema prefix set in RCU while crating BI schema>
EXISTING_DB_PASSWORD=<DB password>
PORT_RANGE_START=7001
PORT_RANGE_END=7999
#Location of static ports file. Only set if PORT_RANGE_START is 'STATIC'
STATIC_PORTS_FILE=
#Select type of initial application. Supported types are
#APPLICATION_TYPE_BI_BUNDLE, APPLICATION_TYPE_EMPTY.
APPLICATION_TYPE=APPLICATION_TYPE_EMPTY
#Location of BI Bundle file to use as the initial application. Only used
if #APPLICATION_TYPE is APPLICATION_TYPE_BI_BUNDLE.
BI_BUNDLE_FILE=
#Password which secures existing BI archive file, change if required
BI_BUNDLE_PASSWORD=<SECURE VALUE>
#New password which will secure AppLite as it is modified with customer
data, #change if required
BI_APP_LITE_PASSWORD=<SECURE VALUE>
BI_DEFAULT_SERVICE_INSTANCE_KEY=<if left blank then default will be set
to SSI>

```

3. If your DOMAINS_DIR is for example: /scratch/Oracle/installation/Middleware/user_projects/domains, then create the same before running config.sh or config.cmd.
4. Change the directory to ORACLE_HOME/bi/bin.
5. Run configuration in silent mode as shown, assuming your response file, oas.resp is available.
 - Linux: ./config.sh -silent -responseFile /scratch/oas.resp
 - Windows: config.cmd -silent -responseFile c:/oas.resp

After you successfully configure Oracle Analytics Server, continue with the Edit Your Input File section. See [Edit Your Input File](#).