

Oracle® Fusion Middleware

Getting Started with SAP ABAP ERP Adapter for Oracle Data Integrator

12c (12.1.3)

E51092-02

January 2015

E51092-02

Copyright © 2010, 2015, Oracle and/or its affiliates. All rights reserved.

Primary Authors: Laura Hofman, Joshua Stanley, Aslam Khan

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

Intel and Intel Xeon 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, Opteron, the AMD logo, and the AMD Opteron 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	v
Audience.....	v
Documentation Accessibility	v
Related Documents	v
Conventions	vi
1 Overview of Oracle Data Integrator SAP ERP Knowledge Modules	
Overview.....	1-1
2 Setting Up the Environment	
Before You Begin.....	2-1
System Requirements and Certification	2-1
Setting Up the SAP System.....	2-3
Gathering SAP Connection Information.....	2-3
Installing Oracle Data Integrator	2-3
Installing and Configuring the Oracle DB Target.....	2-4
Installing and Configuring JCo.....	2-4
Configuring Oracle Data Integrator	2-5
Add the Open Tool	2-5
3 Configuring the Topology	
Configuring the File Server for SAP ABAP	3-1
Configuring the Target Data Server	3-2
Configuring the SAP Source Server	3-3
Configuring the Data Server.....	3-4
Configuring the Logical Schema.....	3-5
4 Creating a New Project	
Creating a New Project.....	4-1
5 Reverse-Engineering Data Models	
Reverse-Engineering the Oracle Target.....	5-1
Reverse-Engineering the SAP Source Datastores	5-2
Creating the Data Model.....	5-2

Validating the SAP Connection	5-3
Starting the Reverse-Engineering Process.....	5-5
Using the SAP Metadata Browser	5-5
The Tree View.....	5-6
The Search Panel	5-7
Reverse-Engineering Without GUI.....	5-8

6 Creating and Running the Mapping

Creating the Mapping	6-1
Create the Mapping	6-1
Define the Source and Target Datastores	6-1
Define Joins between Sources.....	6-2
Create the Mappings.....	6-3
Define the Mapping Flow	6-4
Running the Mapping	6-4
Running the Mapping	6-4
Review the Mapping Execution.....	6-5
Review the Resulting Data.....	6-5

7 Going Further with Oracle Data Integrator

Going further with Oracle Data Integrator	7-1
--	------------

8 More Information

Preface

The purpose of this document is to provide you with an overview of the possibilities offered by SAP ERP Knowledge Modules and how to use them in Oracle Data Integrator.

This document provides examples to help you understand how to achieve an ETL project with Oracle Data Integrator, loading a Data Warehouse with data from SAP systems, while validating data quality.

This project should take approximately 90 minutes to complete. You may save your work in order to pause and resume at any point in the project.

It is recommended that you use Oracle Data Integrator before proceeding with this guide. A good introduction to Oracle Data Integrator is the guide *Oracle Fusion Middleware Getting Started with Oracle Data Integrator*.

It is recommended that you also review the SAP ABAP chapter of the *Developing Knowledge Modules with Oracle Data Integrator* for more information about the SAP ERP KMs.

Audience

This document is intended for developers and administrators who want to use Oracle Data Integrator Application Adapters with SAP ERP systems using SAP JCo libraries, as a development tool for their integration processes.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following documents in *Oracle Data Integrator Library*.

- Release Notes for Oracle Data Integrator

- Understanding Oracle Data Integrator
- Administering Oracle Data Integrator
- Developing Integration Projects with Oracle Data Integrator
- Installing and Configuring Oracle Data Integrator
- Upgrading Oracle Data Integrator
- Application Adapters Guide for Oracle Data Integrator
- Developing Knowledge Modules with Oracle Data Integrator
- Connectivity and Knowledge Modules Guide for Oracle Data Integrator
- Migrating From Oracle Warehouse Builder to Oracle Data Integrator
- Oracle Data Integrator Tool Reference
- Data Services Java API Reference for Oracle Data Integrator
- Open Tools Java API Reference for Oracle Data Integrator
- Getting Started with SAP ABAP BW Adapter for Oracle Data Integrator
- Java API Reference for Oracle Data Integrator
- *Oracle Data Integrator 12c Online Help*, which is available in ODI Studio through the JDeveloper Help Center when you press **F1** or from the main menu by selecting **Help**, and then **Search** or **Table of Contents**.

Conventions

The following text conventions are used in this document:

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

Overview of Oracle Data Integrator SAP ERP Knowledge Modules

This chapter provides an overview of Oracle Data Integrator SAP ERP Knowledge Modules.

This chapter includes the following sections:

- [Overview](#)

Overview

The Oracle Data Integrator SAP ABAP Knowledge Modules provide integration from SAP ERP systems using SAP JCo libraries. This set of KMs has the following features:

- Reads SAP data from SAP ERP system
- Loads this SAP data into Oracle or non-Oracle Staging Area
- Reverse-engineers SAP metadata and proposes a tree browser to pick up only the required metadata

This adapter includes three knowledge modules:

- **RKM SAP ERP:** Its main role is to perform customized reverse-engineering of SAP tables into ODI datastores (Models). The SAP ERP RKM is in charge of connecting to the SAP System, which provides the SAP table's metadata information, and transforming and writing the resulting metadata into Oracle Data Integrator's repository.
- The **LKM SAP ERP to Oracle (SQLLDR)** connects to SAP and retrieves data from the SAP tables defined in the SAP Model. It is used in the mappings. The LKM is in charge of loading source data from a SAP System to an Oracle staging area.
- The **LKM SAP ERP to SQL** connects to SAP and retrieves data from the SAP tables defined in the SAP Model. It is used in the mappings. The LKM is in charge of loading source data from a SAP System to a non-Oracle staging area.

Setting Up the Environment

This chapter describes how to set up your SAP ERP environment for use with Oracle Data Integrator.

This chapter includes the following sections:

- [Before You Begin](#)
- [Installing Oracle Data Integrator](#)
- [Installing and Configuring the Oracle DB Target](#)
- [Installing and Configuring JCo](#)
- [Configuring Oracle Data Integrator](#)

Before You Begin

Before you begin, consider the following:

- [System Requirements and Certification](#)
- [Setting Up the SAP System](#)
- [Gathering SAP Connection Information](#)

Caution: All three sub-sections below must be followed and completed successfully before any parts of the ODI SAP Adapter can be used or tested.

Please take into account that these SAP related tasks usually take some time, as they need to be scheduled with your SAP administration team.

All these activities must be completed before proceeding to the ODI related installation activities.

System Requirements and Certification

Before performing any installation you should read the system requirements and certification documentation to ensure that your environment meets the minimum installation requirements for the products you are installing.

The Oracle Data Integrator requirements are listed in the *Installing and Configuring Oracle Data Integrator*.

The requirements specific to the Oracle Data Integrator SAP ABAP Adapter are:

- Oracle Data Integrator 11.1.1.7.0 with Patch 20235601 (ODI SAP Adapter v37) or later ODI 11g version, or Oracle Data Integrator 12.1.3.0.0 with Patch 19561419 (ODI SAP Adapter v37) or later ODI 12c version.

Note: Most of this documentation also applies to the SAP connectivity in Oracle BI-Applications, which requires Oracle Data Integrator 10g, as specified in more detail in *System Requirements and Supported Platforms for Oracle Business Intelligence Applications for SAP*.

- A Java Connector (JCo) version compatible with adapter must be used. The list of supported JCo versions is available in the Certification Matrix available from Oracle Technology Network (OTN). A minimum version of JCo 3.0.2 is required.
- A JVM version compatible with both Oracle Data Integrator and JCo must be used. A minimum version of JVM 1.5 is required due to JCo pre-requisites.
- The connector supports two transfer modes for transferring data from SAP system to the ODI agent: data transfer using a Shared Directory and data transfer through FTP. For details and restrictions see "File Transfer Considerations" of *Application Adapters Guide for Oracle Data Integrator*. Depending on the chosen file transfer mode the following requirements must be met:
 - **Data transfer through a Shared Directory (recommended transfer method)**
The LKM SAP ERP to Oracle (SQLLDR) requires a folder that is shared between the SAP system and the ODI agent. The SAP application server transfers the data by writing it out into a folder that is accessible from the SAP system and the ODI agent machine. This is typically done by sharing a folder of the ODI agent machine with the SAP system. Note that the shared folder does not necessarily have to be located on the ODI agent machine. A shared folder on a third machine is also possible, as long as the shared folder is accessible to both the ODI agent machine and the SAP system.

Note: For security reasons, the SAP folders should not be shared with the ODI agent. Share only folders of the ODI agent machine with the SAP system.

The shared folder must be accessible to SAP system and not just to the underlying operating system. This means that the folder needs to be declared in SAP transaction AL11 and the folder opens successfully in AL11.

- **Data transfer through FTP** LKM SAP ERP to Oracle (SQLLDR) requires a FTP server to upload data from the SAP ERP system. This data is either read locally by the agent executing the mapping (when this agent runs on the FTP server machine), or remotely (when this agent is located on a different machine than the FTP server). This FTP server must be accessible over the network from both the SAP ERP machine and the agent machine.
- For LKM SAP ERP to Oracle (SQLLDR) only: SQL*Loader is required on the machine running the agent the executed mappings using LKM SAP ERP to Oracle (SQLLDR). SQL*Loader is used for loading data extracted from SAP to the Oracle staging area.

Setting Up the SAP System

The SAP Adapter integrates with the SAP system through some SAP RFCs. These components need to be installed by the SAP Administration team. Please ask your SAP administrator to perform the installation steps described in section *Installing ODI SAP Components* and then the validation steps described in section *Validating the ODI SAP Setup* of the *Application Adapters Guide for Oracle Data Integrator*.

Gathering SAP Connection Information

In order to connect to the SAP ERP system, you must request the following information from your SAP administrators:

- **SAP ERP System IP Address or Hostname:** IP address/ Hostname is the technical name given to the host on which SAP is running.
- **SAP User:** SAP User is the unique user name given to a user for logging on the SAP System.
- **SAP Password:** Case-sensitive password used by the user to log in.
- **SAP Language:** Code of the language used when logging in For example: EN for English, DE for German.
- **SAP Client Number:** The three-digit number assigned to the self-contained unit which is called *Client* in SAP. A Client can be a training, development, testing or production client or represent different divisions in a large company.
- **SAP System Number:** The two-digit number assigned to a SAP instance which is also called Web Application Server or WAS.
- **SAP System ID:** The three-character, unique identifier of a SAP system in a landscape.
- **SAP Transport Layer Name:** This string uniquely identifies a transport layer in a SAP landscape. It allows ODI to create transport requests for later deployment in SAP. Even though there is a default value here, this transport layer name **must** be provided by your SAP Basis team. Not doing so may result in significant delays during installation.
- **SAP Version:** The version of the SAP system.
- **SAP Character Set:** The character set is only required if your SAP system is not a UNICODE system. For a complete list of character sets, see "Locale Data" in the Oracle Database Globalization Support Guide. For example, EE8ISO8859P2 for Croatian Data. For UNICODE systems, use UTF8.

Note: All the connection data listed above (except SAP SNC Connection Properties and SAP Router String) are mandatory and should be requested from the SAP Administrators. You may consider requesting support during connection setup from your SAP administrators.

Installing Oracle Data Integrator

Before starting with this project, you need to install and configure Oracle Data Integrator. See the *Oracle Fusion Middleware Getting Started with Oracle Data Integrator* and the *Installing and Configuring Oracle Data Integrator* for more information.

Installing and Configuring the Oracle DB Target

This project uses an Oracle database engine as the target and as the staging area for the mappings. You can download for example Oracle Database 11g Express Edition from Oracle Technology Network (OTN). Install and configure this database.

This project targets an Oracle table that can be created using the following script:

```
-- Create demo target schema
CREATE USER ODI_SAP_DEMO IDENTIFIED BY ODI_SAP_DEMO DEFAULT TABLESPACE USERS
TEMPORARY TABLESPACE TEMP;
GRANT CONNECT, RESOURCE TO ODI_SAP_DEMO;

-- Create demo target table
CREATE TABLE ODI_SAP_DEMO.W_GEO_DS (
    LANGUAGE_KEY CHAR(20),
    COUNTRY VARCHAR(50),
    COUNTY VARCHAR(50),
    STATE_PROV VARCHAR(50),
    NATIONALITY VARCHAR(50));

ALTER TABLE ODI_SAP_DEMO.W_GEO_DS ADD CONSTRAINT PK_W_GEO_DS PRIMARY KEY
(LANGUAGE_KEY);
```

Installing and Configuring JCo

The SAP adapter uses Java Connector (JCo) to connect to the SAP system. JCo must be configured before proceeding with the project.

To install and configure JCo:

1. Download a supported JCo version for your configuration from <http://service.sap.com/connectors>. Check the supported JCo version in the Certification Matrix available at Oracle Technology Network. Note that a minimum version of JCo 3.0.2 is required.

Notes:

- Choose the SAP JCo package matching your operating system and your system architecture (32/64Bit). E.g. if you are running ODI inside a 32-Bit JVM, you must download the 32-Bit SAP JCo, even if the CPU and OS are 64-Bit. Mixing 32-bit and 64-bit architecture is not possible due to native libraries required by SAP JCo and will result in connection failure.
- odi.conf contains the JDK path used for ODI Studio.

2. Unzip the appropriate distribution package into an arbitrary directory <sapjco-install-path>.
3. Follow the installation instructions in <sapjco-install-path>/javadoc/installation.html for the respective platform.
4. Copy sapjco3.jar and sapjco3.dll:

For ODI 10g: Copy sapjco3.jar and sapjco3.dll (or respective binary like libsapjco3.so) into the oracledi/drivers directory.

For ODI 11g: Copy sapjco3.jar and sapjco3.dll (or respective binary like libsapjco3.so) into the <ODI_HOME>/odi_misc directory (ODI Studio) and into the <ODI_HOME>/oracledi/agent/drivers directory (Standalone Agent) and into the <WLS_DOMAIN>/lib directory (JEE Agent).

For ODI 12c: Copy sapjco3.jar and sapjco3.dll (or respective binary like libsapjco3.so) into the <ODI_HOME>/odi/sdk/lib directory (ODI Studio, ODI Standalone Agent) and into the <WLS_DOMAIN>/lib directory (JEE Agent).

5. Restart ODI Studio and any agents.
6. Check the JCo installation. This will be also checked later in this guide.

Configuring Oracle Data Integrator

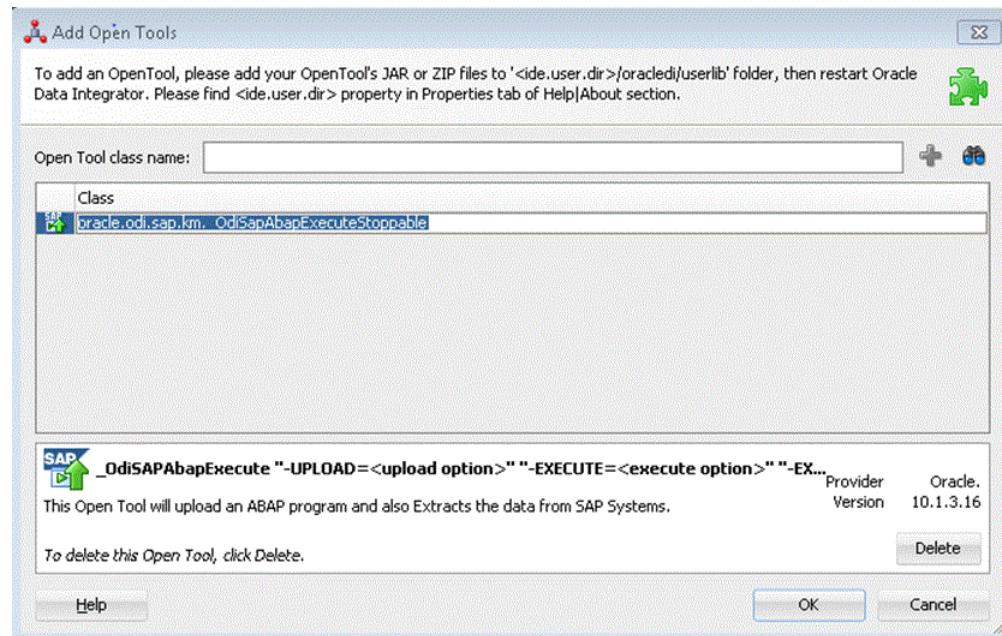
Perform the following steps to configure Oracle Data Integrator:

1. [Add the Open Tool](#)

Add the Open Tool

The complete process of installing and adding an Open Tool to ODI is described in *Oracle Data Integrator Tool Reference*. This section details only the SAP ABAP specific steps.

1. Connect to Designer.
2. Select **File > Add/Remove Open Tools...**
3. In the Add/remove Open Tools window, enter the following name in the Open Tool class name field:
`oracle.odi.sap.km._OdiSapAbapExecuteStoppable`
4. Click **Add Open Tool**.
5. The Add Open Tools window appears as shown below.



6. Click **OK**.

Configuring the Topology

This chapter describes how to configure the topology by using a practical example.

This project will integrate the following source and target data:

Source tables *T005T* and *T005U* are the tables from SAP containing geographical information.

Target *W_GEO_DS* is a dimension table in an Oracle database that needs to be loaded from the SAP Sources. This table contains geographical information such as Continent name, Country name, Region, State, County, City and Zip code.

This chapter includes the following sections:

- [Configuring the File Server for SAP ABAP](#)
- [Configuring the Target Data Server](#)
- [Configuring the SAP Source Server](#)

Configuring the File Server for SAP ABAP

Perform the following operations after installing or upgrading your Oracle Data Integrator version:

1. Connect to Topology Manager.
2. If this SAP ABAP technology does not exist in your Master Repository, import the SAP ABAP technology in Synonym INSERT_UPDATE mode from the `/impexp` folder.
3. Perform an upgrade of the Master Repository. Refer to the *Installing and Configuring Oracle Data Integrator* for more information on the Master Repository upgrade process.
4. In Topology Manager, open the JavaBeanShell technology and check on the Language tab that the JYTHON language is listed. If not, add it.
5. Create a File data server pointing to an existing FTP/file server into which the extraction file will be pushed from SAP and picked up for SQL*Loader. For use with FTP transfer, set the following parameters. For use with shared folder transfer, the settings on this data server do not matter.
 - **Host (Data Server):** FTP server IP host name or IP address.
 - **User:** Username to log into FTP server.
 - **Password:** Password for the user.

6. For use with "LKM SAP ERP to SQL" these additional parameters must be configured:
 - JDBC driver class: com.sunopsis.jdbc.driver.file.FileDriver
 - JDBC URL: jdbc:snps:dbfile?ENCODING=UTF8

The above URL is for SAP UNICODE systems. For non-UNICODE systems, please see details on ENCODING parameter in "Creating a File Data Server" of *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*. The encoding chosen on this URL must match the code page used by the SAP Application Server.
7. In this File data server create a Physical Schema representing the folder in the File server to which the extraction file will be pushed. For use with FTP transfer, see details below. For use with shared folder transfer and for more details, see "Transfer using a Shared Directory (recommended)" of *Application Adapters Guide for Oracle Data Integrator*.
 - **Data Schema:** Path on the FTP server to upload or download extraction files from the remote location. This path is used when uploading extraction files from the SAP ERP system into the FTP server. It is also used by a remote agent to download the extraction files. Note that this path must use slashes and must end with a slash character.
 - **Work Schema:** Local path on the FTP server's machine. This path is used by an agent installed on this machine to access the extraction files without passing via the FTP server. This access method is used if the `FTP_TRANSFER_METHOD` parameter of the LKM SAP ERP to Oracle (SQLLDR)/ LKM SAP ERP to SQL is set to NONE. As Work Schema is an OS file name, slashes/ backslashes should be used according to OS.

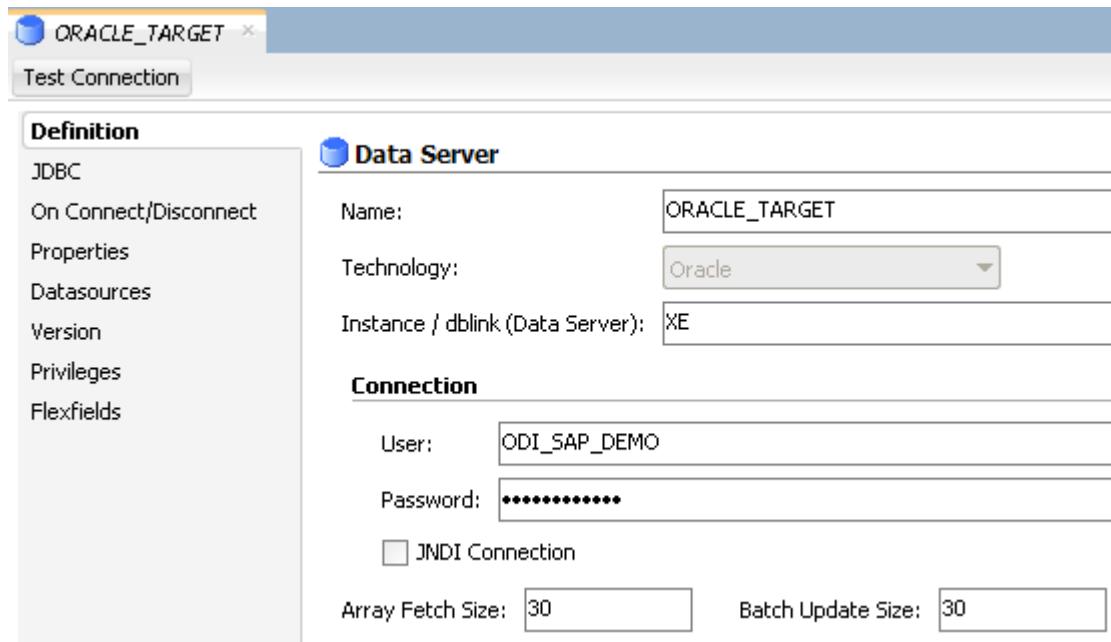
Path names need to end on slash/ backslash. Path names given on Data and Work schemas are not necessarily the same: the FTP service may provide access to a FTP directory named `/sapfiles` while the files can be stored locally in `c:\inetpub\ftproot\sapfiles`.

Refer to "File Transfer Considerations" of *Application Adapters Guide for Oracle Data Integrator*.
8. If the corresponding Logical Schema called `File Server for SAP ABAP` does not exist yet, create it. This Logical Schema name is fixed and must be mapped to the Physical Schema created in the previous step.

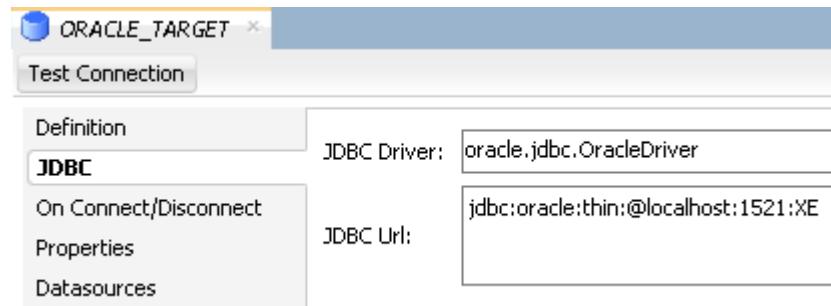
Configuring the Target Data Server

This section describes how to setup an Oracle Data Server. Please use respective settings when configuring a non-Oracle Target Data Server for use with LKM SAP ERP to SQL.

1. Connect to Topology Manager.
2. In the Physical Architecture tree view, expand the Technologies node and select the Oracle technology.
3. Right-click and select **New Data Server**.
4. Enter the data server Definition as shown below:



5. Select the JDBC tab, and enter the JDBC connection information to your data server as shown below:



Make sure to change the URL to match your Oracle instance configuration.

6. Click **Test Connection** to test the connection to this data server.
7. From **File** menu, click **Save**.
8. Right-click the data server you created and select **New Physical Schema**.
9. In the Physical Schema windows that appears, select the **ODI_SAP_DEMO** for both Schema and Work Schema.

If you already have a Work Schema for ODI defined for this data server, you can use it instead of the **ODI_SAP_DEMO** Schema.

10. Go to the Context tab.
11. Click **Add** and enter in the Logical Schema field the value **ODI_SAP_DEMO**.
12. From the **File** menu, click **Save**.

Configuring the SAP Source Server

The following topics describe how to configure the SAP source server:

- [Configuring the Data Server](#)
- [Configuring the Logical Schema](#)

Configuring the Data Server

Except for the data server name, all the parameters that you provide while defining the SAP Data server should be provided by the SAP Administrators. See ["Setting Up the SAP System"](#) on page 2-3 for more information.

The SAP Adapter requires privileges to perform set up and execution operations. Please provide your administrators with the list of privileges listed in *B1. SAP ABAP ERP Required Privileges in Application Adapters Guide for Oracle Data Integrator*. These privileges are required for the SAP user that they will provide you to login the SAP System.

See ["Gathering SAP Connection Information"](#) on page 2-3 for more information about these parameters.

1. In the Physical Architecture tree view, expand the Technologies node, select the SAP ABAP technology.
2. Right-click and select **New Data Server**.
3. Enter the data server definition. Set the parameters for this data server as follows:
 - **Name:** SAP_ERP. The name of the data server as it will appear in ODI.
 - **Host (Data Server):** SAP ERP System IP Address or Hostname.
 - **User:** SAP User, as provided by the SAP Administrator.
 - **Password:** This user's SAP Password. This password is case-sensitive.
4. Set the Flexfield values for this data server in the Flexfields tab:
 - **SAP Language:** Code of the language used when logging in. For example EN for English, DE for German.
 - **SAP Client Number:** The three-digit number assigned to the self-contained unit which is called *Client* in SAP. A Client can be a training, development, testing or production client or represent different divisions in a large company.
 - **SAP System Number:** The two-digit number assigned to a SAP instance which is also called Web Application Server or WAS.
 - **SAP System ID:** The three-character, unique identifier of a SAP system in a landscape.
 - **SAP Character Set:** The character set is only required if your SAP system is not a UNICODE system. For a complete list of character sets, see "Locale Data" in the Oracle Database Globalization Support Guide. For example, EE8ISO8859P2 for Croatian Data. For UNICODE systems, use UTF8.

Note: This FlexField is only used with "LKM SAP to Oracle (SQLLDR)". For use with "LKM SAP to SQL" please define the character set using the ENCODING parameter in the file driver URL, as described in ["Configuring the File Server for SAP ABAP"](#) on page 3-1 for non-UNICODE SAP systems.

- **SAP ABAP Version:** Enter the SAP ABAP version as follows:

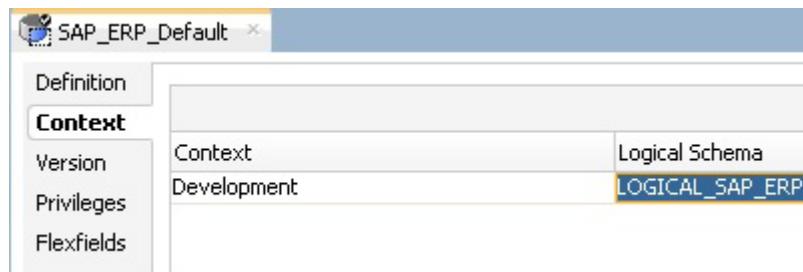
- For SAP 4.6C enter 46C
- For SAP 4.7 enter 620
- For SAP ECC 5.0 enter 640
- For SAP ECC 6.0 enter 700
- **SAP BW Version & SAP ERP Version:** unused in SAP ERP Connector.
- **SAP Transport Layer Name:** This transport layer is used by ODI to create any new transport requests. The default transport layer name is SAP. If the SAP system uses a different transport layer, this FlexField must be updated accordingly. Otherwise, any TR creation will fail.

5. From the **File** menu, click **Save**.

Note: The **Test Connection** button for validating SAP Connection definition is not supported for this connection.

Configuring the Logical Schema

1. In the Physical Schema window, do not edit the Definition tab.
2. Select the Context tab, click **Add** and enter the Logical Schema name **LOGICAL_SAP_ERP**.



3. From the **File** menu, click **Save**.

4

Creating a New Project

This chapter describes how to create a new Project for SAP.

In order to work with the data servers you have created, you must create a project and import the appropriate knowledge modules for reverse-engineering the data structures and integrating the data.

Creating a New Project

To create a new Project for SAP:

1. Connect to Designer.
2. In the Projects tree view click **New Project**.
3. In the Definition tab, enter `SAP Demo` for the Project name.
4. From the **File** menu, click **Save** to save your project.
5. Expand the `SAP Demo` Project, select the `First Folder` node.
6. Rename this folder to `SDE_SAP_GeoDimension`.
7. Right-click the `SAP Demo` Project in the Projects tree view.
8. Right-click and select **Import > Import Knowledge Modules**.
9. Select in the File import directory the directory containing your KMs. By default, they are located in the `../impexp` directory.
10. Select the following KMs:
 - IKM Oracle Incremental Update
 - LKM SAP ERP to Oracle (SQLLDR)
 - LKM SAP ERP to SQL
 - RKM SAP ERP
 - RKM SAP ERP Connection Test
11. Click **OK** to perform the import.

Reverse-Engineering Data Models

This chapter describes how to reverse-engineer the Oracle target and SAP source data models.

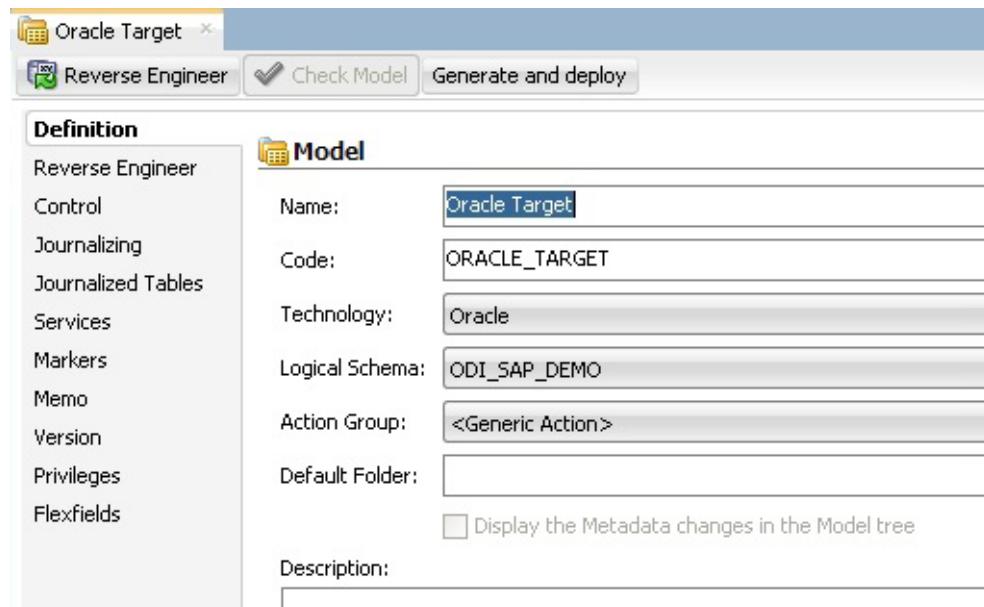
This chapter includes the following sections:

- [Reverse-Engineering the Oracle Target](#)
- [Reverse-Engineering the SAP Source Datastores](#)

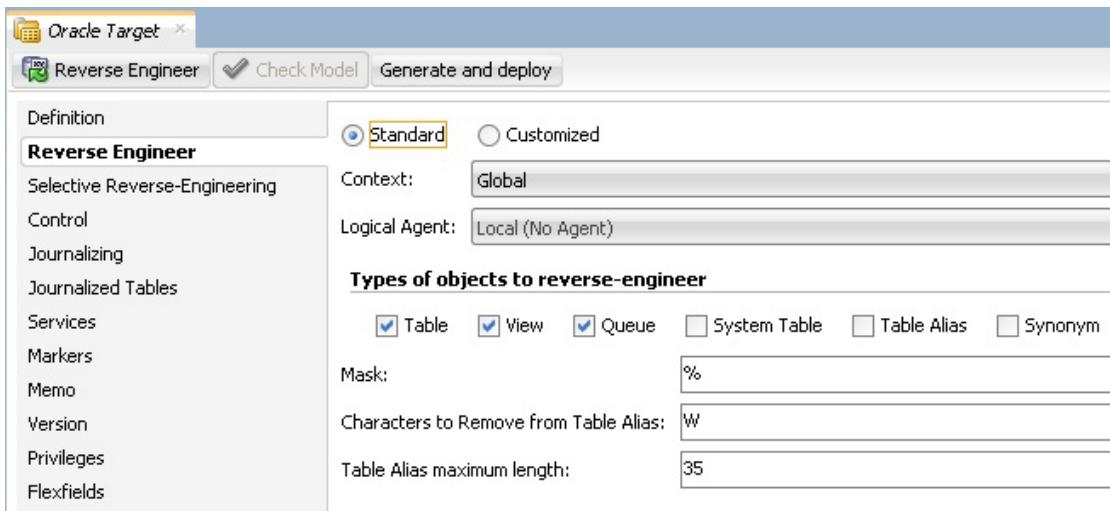
Reverse-Engineering the Oracle Target

To reverse-engineer the Oracle data model:

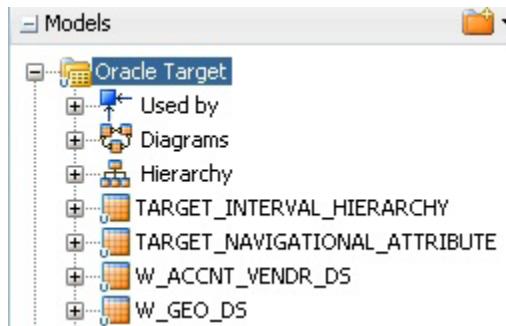
1. Connect to Designer.
2. In the Models tree view, click **New Model**.
3. In the Definition tab, enter the model parameters as shown below:



4. In the Reverse Engineer tab, select the **Global** context.



5. Click **Reverse Engineer**, and then save your model.
6. The WS_GEO_DS datastore appears in your Oracle model, as shown below.



Reverse-Engineering the SAP Source Datastores

To reverse-engineer the SAP Source datastores you need to perform the following tasks:

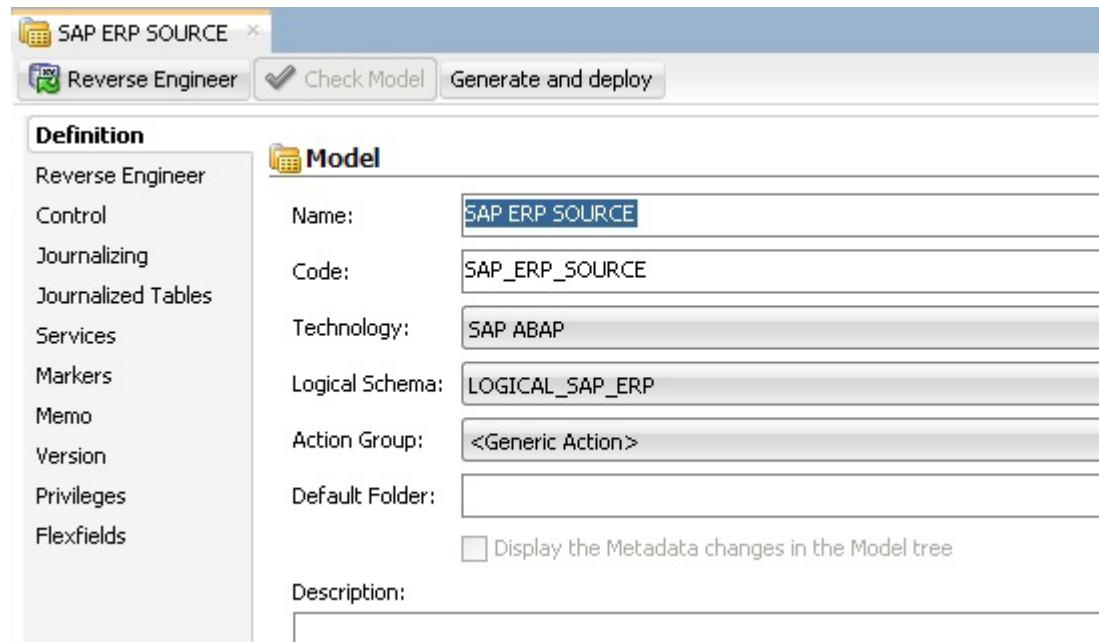
1. [Creating the Data Model](#)
2. [Validating the SAP Connection](#)
3. [Starting the Reverse-Engineering Process](#)

This section also includes the following topics:

- [Using the SAP Metadata Browser](#)
- [Reverse-Engineering Without GUI](#)

Creating the Data Model

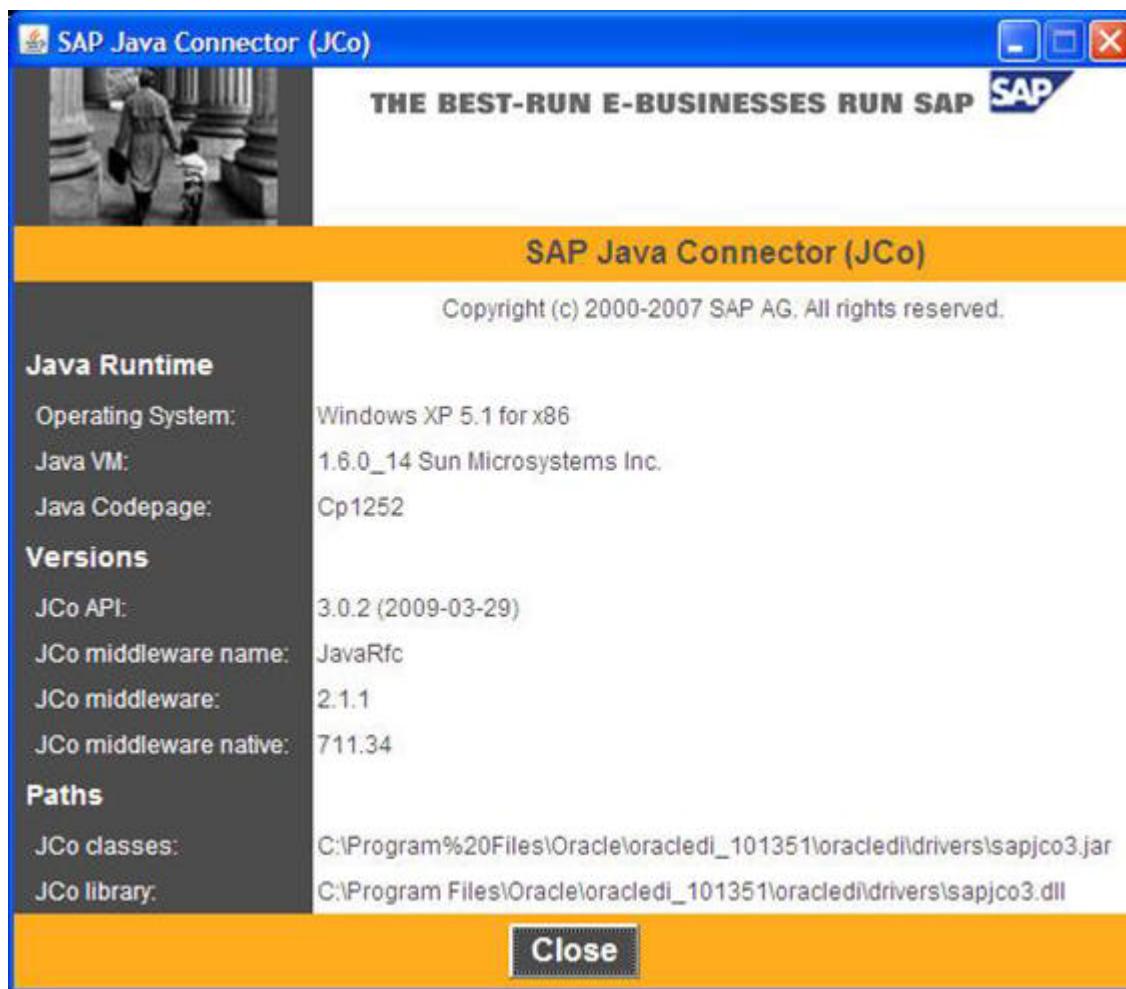
1. In the Models tree view, click **New Model**.
2. In the Definition tab, enter the model parameters as shown below. You must select the SAP ABAP technology and the Logical Schema previously created (in our example it is the LOGICAL_SAP_ERP Logical Schema).



3. In the Reverse Engineer tab:
 - a. Select the Global context.
 - b. Select Customized option.
 - c. Select the RKM SAP ERP Connection Test you have imported in the SAP ERP Demo project.

Validating the SAP Connection

1. In the Models tree view, open the SAP ERP Source Model.
2. Click **Reverse Engineer** and confirm the start of the reverse-engineering process. This process runs two tests:
 - Establish a test connection to the SAP system and
 - Validate the proper setup of SAP JCo by displaying the About dialog.
3. In a few seconds, you should see the SAP JCo-About dialog as shown here:



Please verify the JCo version number (see "System Requirements and Certification" on page 2-1 for more details) and verify that a path for the JCo Library is given (.dll for Windows). If this is not the case or you see any error message in this about dialog, please review the SAP JCo installation, as described in your SAP JCo package, and run this test again.

Note: Closing the JCo-About-Dialog will also close ODI.

4. Once you have validated SAP JCo installation, open ODI Operator.
5. In the Operator, expand the Reverse-Engineering session down to the task level.
6. Verify whether the task "Test SAP Connection" has been executed successfully. If not, view the task details to identify the connection problem. The output of this task *must* contain Successfully connected to SAP System. If not, the connection test has failed and the connection problem must be analyzed *before* you continue.

Do not continue until you have successfully validated the SAP connection!

Note: If you want to validate the SAP connectivity independent of ODI, please see *B3. SAP Stand-Alone Connection Test* in *Application Adapters Guide for Oracle Data Integrator*.

Starting the Reverse-Engineering Process

To start the reverse-engineering process of the SAP ERP datastores:

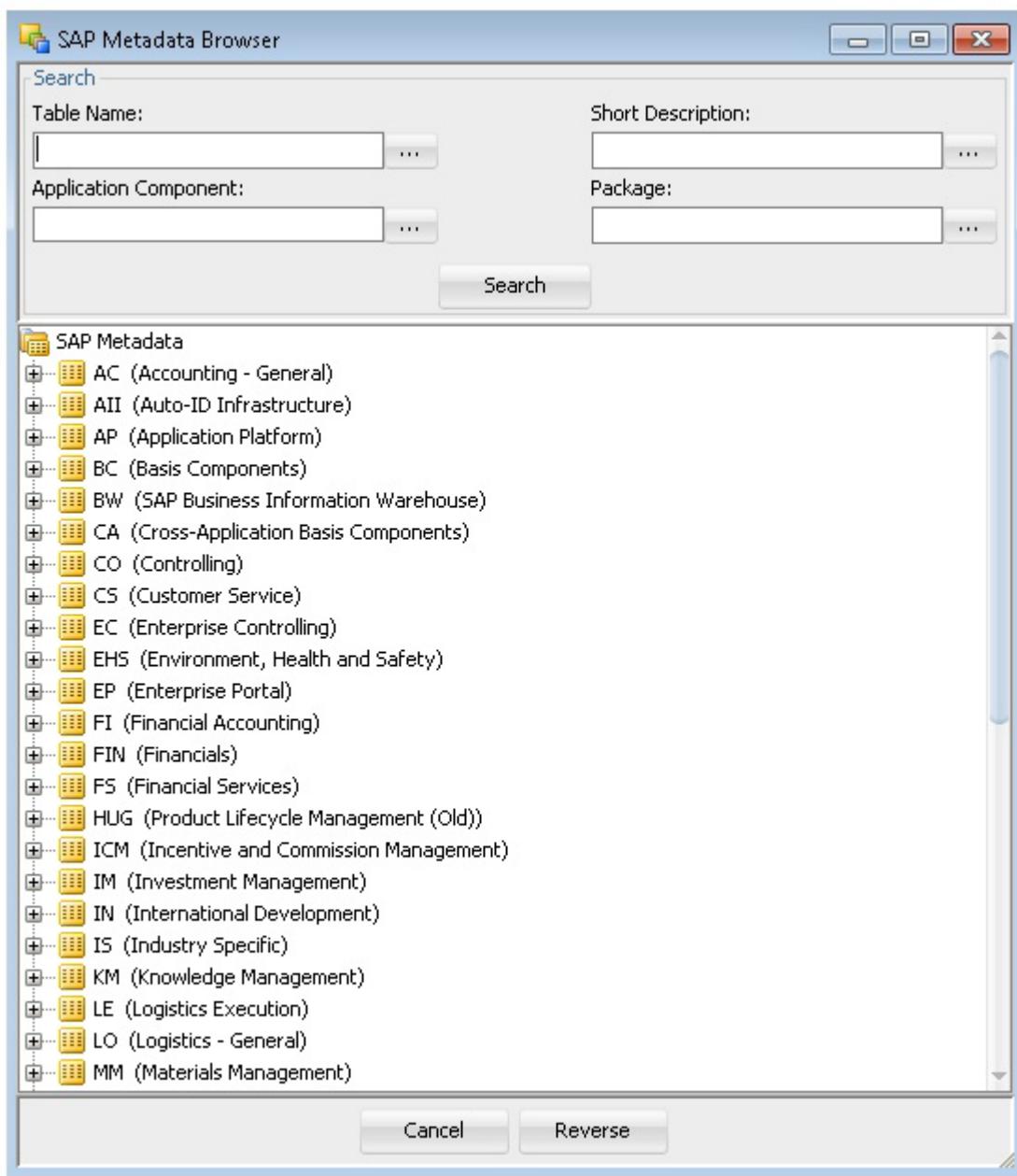
1. In the Models tree view, open the SAP ERP Source Model.
2. In the Reverse Engineer tab:
 - a. Select the Global context.
 - b. Select the Customized option.
 - c. Select the RKM SAP ERP you have imported in the SAP ERP Demo project.
 - d. Set the USE_GUI KM option to Yes.
 - e. Set other parameters according to your SAP configuration. See the *Application Adapters Guide for Oracle Data Integrator* for more information on the RKM options.
3. From the File menu, click **Save** to save your changes.
4. Click **Reverse Engineer** to start the reverse-engineering process.
5. Click **OK**.
6. The Sessions Started Dialog is displayed.
7. Click **OK**.
8. For first time RKM use only: Validate in Operator that the session is now in status running. If session has failed, please validate settings. Do not move on until all installation steps have been completed successfully.

Using the SAP Metadata Browser

As you have set the USE_GUI option to Yes, the RKM displays in the **SAP Metadata Browser** the SAP objects organized in the tree view as shown below.

Note: In case that the below dialog does not show up, see ODI Operator and ODI log messages for details.

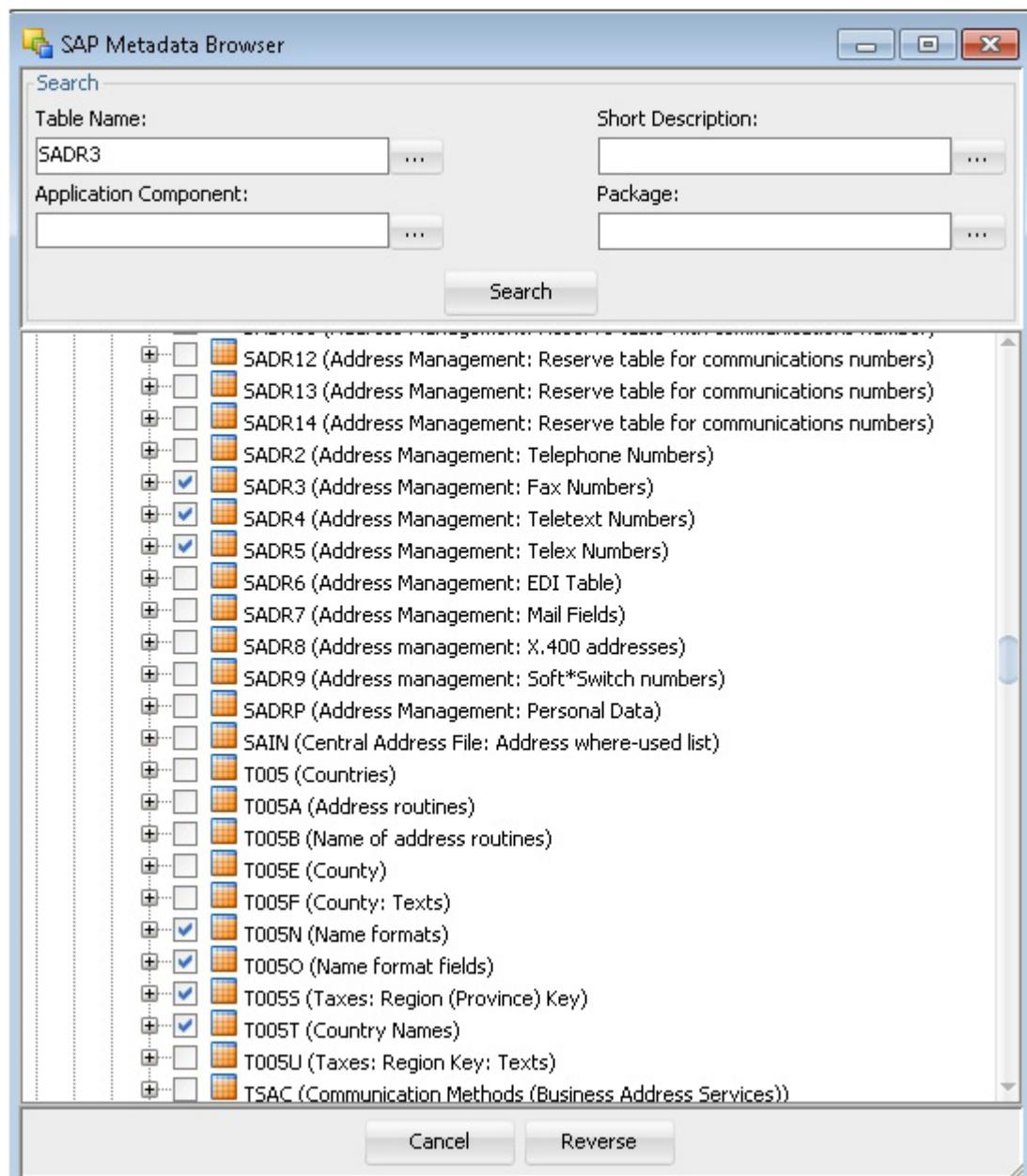
Refer to "Log Files," in *Application Adapters Guide for Oracle Data Integrator* for more details.



The Tree View

The SAP Modules are organized into a tree view. Only the SAP Modules available to the current user are displayed. By expanding the tree view, you can see SAP tables, and select those that you want to reverse-engineer.

Note that only the tables selected in the tree view are reverse-engineered.



The Search Panel

The **Search** option group allows you search specific objects in the browser, based on:

- Table Name
- Short Description
- Application Component
- Package

For the example project, the T005T and T005U country tables are used:

1. Enter T005T in the Table Name field and click **Search** to find the first table to reverse-engineer. Select this table.
2. Enter T005U in the Table Name field and click **Search** to find the second table to reverse-engineer. Select this table.

3. Click **Reverse**.
4. Oracle Data Integrator starts the reverse-engineering process on the selected datastores. The reverse-engineered datastores appear under the Model.

Reverse-Engineering Without GUI

If you want to perform a reverse-engineering without using the SAP Metadata Browser, you can enter the following parameters in the SAP ERP KM options:

- USE_GUI: No
- SAP_TABLES_NAME : T005T, T005U

With this configuration, the tables will be reverse-engineered without displaying the SAP Metadata Browser.

By using the appropriate KM options, you can also select a filter for the reversed tables based on for example a SAP Module or a Package name. You can also specify a table name mask in the **Mask** field of the Reverse tab to filter table names.

Refer to the *Application Adapters Guide for Oracle Data Integrator* for more information on the RKM options.

Note: It is not possible to view the data in the SAP ERP tables from Oracle Data Integrator.

What happens when you reverse-engineer SAP ERP tables?

First, Oracle Data Integrator uploads in the SAP system ABAP code allowing for retrieving the SAP metadata. This RFC is uploaded only if the UPLOAD_ABAP_CODE and UPLOAD_ABAP_BASE options are selected (these options are typically activated only once). The ABAP programs are generated in the SAP function group given in the SAP_FUNCTION_GROUP_ODI_DEV and SAP_FUNCTION_GROUP_ODI_PROD RKM options.

Once this code is set up on SAP, Oracle Data Integrator is able to retrieve the SAP metadata and to display it in the SAP Metadata Browser. Selected metadata is then reverse-engineered from SAP into the Oracle Data Integrator repository. The repository only contains the description of the SAP metadata, and no data at all.

In addition to the table definitions organized and enriched with user-friendly information, the keys and indexes definitions are also retrieved. The model appearing in Oracle Data Integrator shows all the information required for performing integration tasks on the SAP ERP Model.

6

Creating and Running the Mapping

This chapter describes how to create and run the mapping.

This chapter includes the following sections:

- [Creating the Mapping](#)
- [Running the Mapping](#)

Creating the Mapping

Now that the source and target data models are created, it is possible to create a mapping to integrate data from the T005T and T005U SAP tables to the WS_GEO_DS Oracle table.

To create the mapping, perform the following steps:

1. [Create the Mapping](#)
2. [Define the Source and Target Datastores](#)
3. [Define Joins between Sources](#)
4. [Create the Mappings](#)
5. [Define the Mapping Flow](#)

Create the Mapping

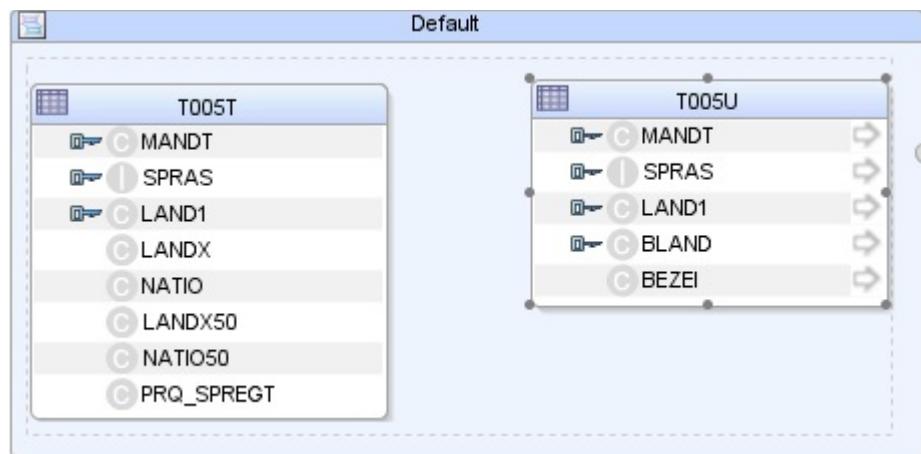
1. Open Designer.
2. In the Projects tree view, expand the SAP Demo Project.
3. Expand the SDE_SAP_GeoDimension folder.
4. Select the Mappings node.
5. Right-click and select **New Mapping**.
6. In the Mapping Definition tab, enter the mapping name: SDE_SAP_GeoDimension.W_GEO_DS.
7. Save the mapping.

Define the Source and Target Datastores

To define the source and target datastores:

- In the Models tree view, select the WS_GEO_DS datastore from the Oracle Target Model.

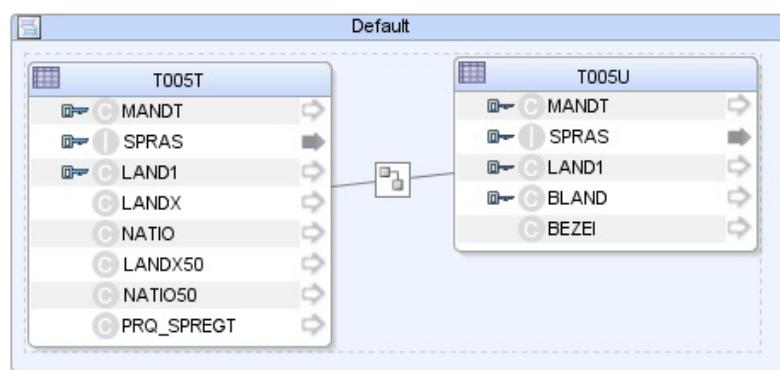
- Drag this datastore into the Target Datastore area of the Logical diagram. (Right area of the Logical diagram)
- In the Models tree view, select the T005T Country Names datastore from the SAP ERP Source model.
- Drag this datastore into the Sources area of the Logical diagram.
- Perform the same operation for the T005U Taxes: Region Key datastore from the SAP ERP Source model. The Sources area of the Logical diagram should look as shown below:



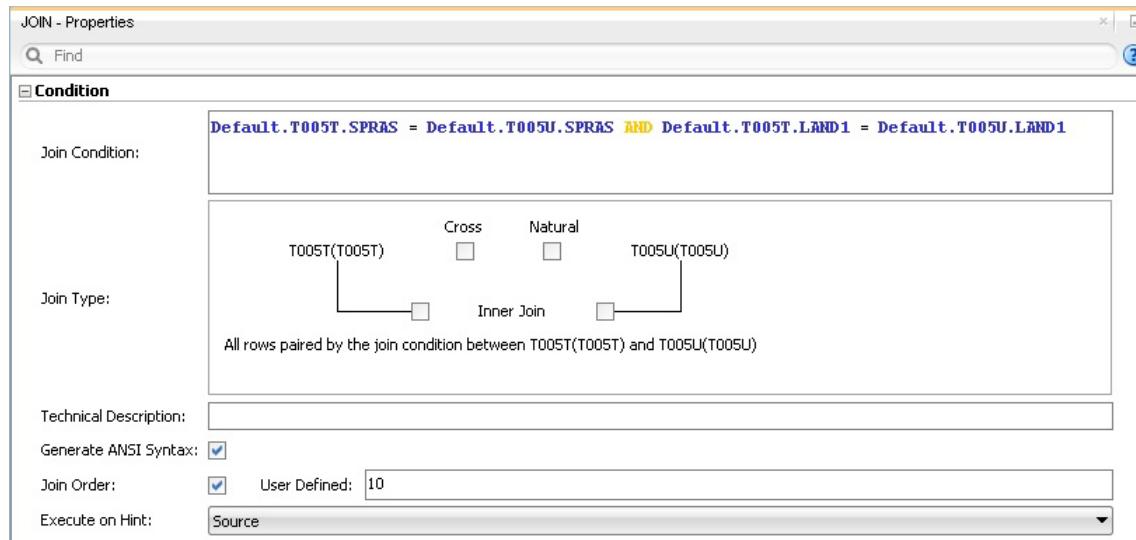
Define Joins between Sources

To create joins between the source datastores of a mapping:

1. In the Sources area of the Logical diagram drag the SPRAS attribute from the T005T Country Name datastore onto the SPRAS attribute in the T005U Taxes: Region Key datastore.
2. On the Create Lookup or Join popup, select **Join** and click **OK**. This defines the first join on the Language Key identifier, as shown below:



3. Drag the LAND1 attribute from the T005T Country Name datastore onto the LAND1 attribute in the T005U Taxes: Region Key datastore. The join is extended with the new attributes. In the Properties panel, you can see the join clause:
`T005T.SPRAS=T005U.SPRAS AND T005T.LAND1=T005U.LAND1`
4. In the Properties panel, select **Generate ANSI Syntax** and **Join Order** options, and make sure that the **Execute on Hint** option is set to **Source** as shown below:

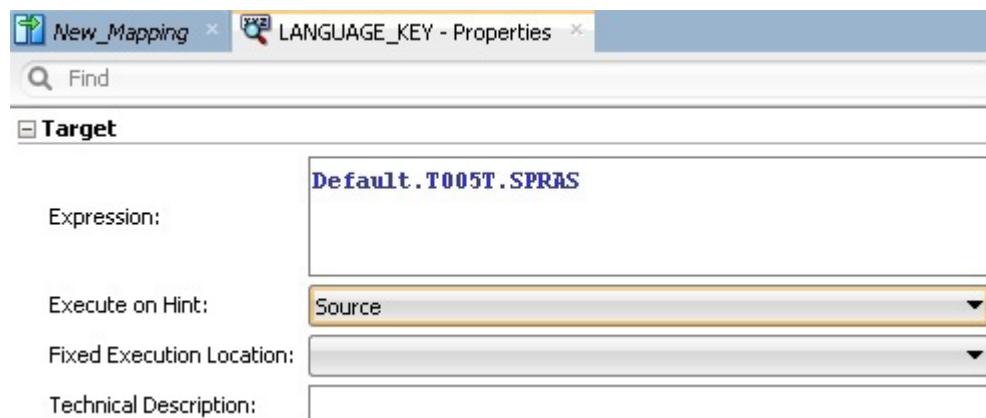


5. Make sure that table number of T005U is smaller than that of T005T. In *1:n relationships* the parent table has to be the first in a join.

Create the Mappings

To create the mappings of the target datastore:

1. Select the LANGUAGE KEY attribute from the target datastore.
2. Drag the SPRAS attribute from the T005T Country Names source datastore onto the LANGUAGE KEY attribute of the target datastore.
3. Make sure that the **Execution on Hint** option is set to **Source**. The mapping should look as shown below:

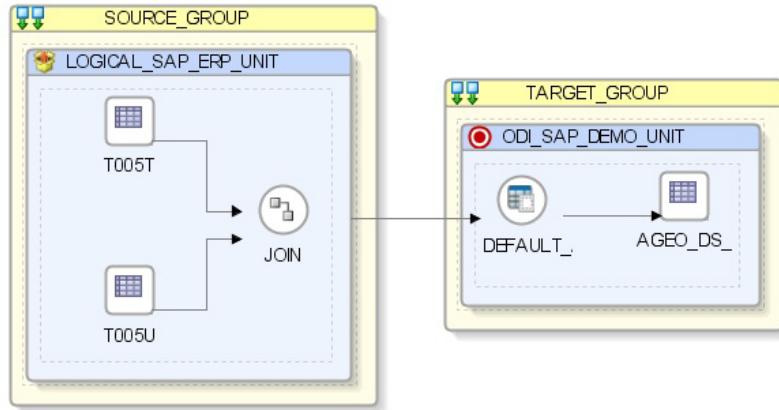


4. Repeat this operation to perform the following simple mappings:
 - W_GEO_DS.COUNTRY = T005T.LAND1
 - W_GEO_DS.COUNTY = T005T.LANDX
 - W_GEO_DS.STATE_PROV = T005U.BLAND
 - W_GEO_DS.NATIONALITY = T005T.NATIO

Define the Mapping Flow

To define the mapping flow:

1. Select the Logical diagram. The Flow diagram looks as shown below;



2. In the target group, select the access point for the source group that contains both the T005T and T005U datastores. This source group represents the source dataset made up of two joined SAP ERP datastores.
3. In the Property panel, under Loading Knowledge Module, select LKM SAP ERP to Oracle (SQLLDR) or LKM SAP ERP to SQL from the Loading Knowledge Module drop-down list.
 - a. If you are using FTP for the data transfer, please set the *FTP_TRANSFER_METHOD* option to FTP. If the FTP server is installed on your ODI agent machine, select NONE.
 - b. Leave other options as they are.
4. In the target group, select the target Oracle database, which is also used as the staging area for this mapping.
5. In Property panel, under Integration Knowledge Module, select IKM Oracle Incremental Update from the Integration Knowledge Module drop-down list.
6. Set the IKM options as follows:
 - a. Select No for the *FLOW_CONTROL* option.
 - b. Leave other options as they are.
7. From the **File** menu, click **Save** and close your mapping.

Running the Mapping

This section contains the following topics:

- [Running the Mapping](#)
- [Review the Mapping Execution](#)
- [Review the Resulting Data](#)

Running the Mapping

To run the mapping:

1. In the Projects tree view, expand the SAP Demo Project
2. Expand the SDE_SAP_GeoDimension folder.
3. Expand the Mappings node.
4. Select the SDE_SAP_GeoDimension.W_GEO_DS mapping.
5. Right-click and select **Execute**.
6. In the Execution window that appears, click **OK**.
7. Click **OK** in the Session Started window.

What happens when you run the mapping?

First, Oracle Data Integrator creates an ABAP program that does the following:

- It performs the extraction of the data on the SAP engine. In this example, the extraction joins the two source tables and then returns the joined resultset in an extraction file.
- It uploads the extraction file into a FTP data server. This FTP host is specified via the File Logical Schema called File Server for SAP ABAP. This Logical Schema is mapped to a Physical Schema and therefore a data server in the given context. This data server contains the connection information for the FTP host.

Then, the ABAP code is uploaded using the *OdiSapAbapExecute* tool. The ABAP program is pushed into the SAP Function group given in the SAP_FUNCTION_GROUP LKM option. This phase can be suppressed by setting the UPLOAD_ABAP_CODE LKM option to No.

The ABAP code is executed also by the *OdiSapAbapExecute* tool. At the end of the ABAP code execution, the extraction file is available in the FTP host.

The Oracle Data Integrator agent is able to download this extraction file from the FTP host, or directly access it, depending on the FTP_TRANSFER_METHOD specified in the IKM option.

Finally, when the agent accesses the extraction file, it uses SQL*Loader or a JDBC Connection to load this file into the Oracle/non-Oracle staging area. The rest of the integration process takes place within the Oracle/non-Oracle engine.

Review the Mapping Execution

To review the mapping execution:

1. Connect to Operator.
2. In the Operator, select the Session List tree view.
3. Expand the All Executions node in this tree view.
4. The latest session is the first entry of this list.
5. Double click the SDE_SAP_GeoDimension.W_GEO_DS node to see the session details and the number of lines processed.

Review the Resulting Data

To review the resulting data:

1. In the Designer, in the Models tree view, expand the Oracle Target Model.

2. Select the `W_GEO_DS` datastore, right-click and select **Data** to view the data integrated into the target Oracle table.

Going Further with Oracle Data Integrator

This chapter describes what steps to take next with Oracle Data Integrator.

You have now completed a project extracting data from a SAP ERP system!

In this project, you have:

- Set up the environment and topology to work with SAP ERP.
- Created and reverse-engineered a SAP ERP data model
- Created a mapping to load the data from several joined SAP ERP tables into the Oracle database.

Note: In case the execution did not complete successfully, please check the ODI Operator and ODI Logs for details. In addition to this, the following log files will contain execution information.

- <System Temp Dir or local FTP dir>/ ZODI_<Mapping Id>_<SrcSet>_<Context>.out
- <System Temp Dir or local FTP dir>/ ZODI_<Mapping Id>_<SrcSet>_<Context>.err

Refer to "Log Files," in *Application Adapters Guide for Oracle Data Integrator* for more details.

Refer to the individual KM options and their descriptions as well as to "Considerations for SAP ERP Integration" in *Application Adapters Guide for Oracle Data Integrator* for more information on the SAP ERP KM features and options.

Going further with Oracle Data Integrator

Use the demonstration environment to familiarize yourself with Oracle Data Integrator. You can go further with Oracle Data Integrator by taking advantage of the samples available on the Oracle Technology Network.

8

More Information

This chapter provides links to sections where you can find more information about the ODI Adapter for SAP ABAP ERP.

More information about the ODI adapter for SAP ABAP ERP can be in the following sections of the *Application Adapters Guide for Oracle Data Integrator*:

- *SAP ABAP ERP*
- *Additional Information for SAP ABAP ERP Adapter*
- *Installing ODI SAP Components*
- *Moving ODI and SAP Components from Development to Production*

