

Oracle Data Integrator

Getting Started with SAP ABAP ERP
Adapter

10g Release 3 (10.1.3)

July 2011

Oracle Data Integrator: Getting Started with SAP ABAP ERP Adapter, 10g Release 3 (10.1.3)

Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

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

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software 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 which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on 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. 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.

Table of Contents

Introduction.....	5
Overview.....	6
Setting up the Environment	7
Before You Begin.....	7
System Requirements and Certification.....	7
Getting the Right Privileges.....	8
Gathering SAP Connection Information.....	8
Validating the Shared Folder Setup.....	8
Validating the FTP Setup	9
Validating SAP Privileges	10
Validating SAP Transport Layer Name.....	10
Installing Oracle Data Integrator	11
Installing and Configuring the Oracle DB Target.....	11
Installing and Configuring JCo.....	12
Setting up an FTP Server	12
Configuring Oracle Data Integrator.....	12
Set up the Topology	12
Add the Open Tool	13
Configuring the Topology	15
Configuring the Oracle Data Server	15
Configuring the SAP Source Server	16
Configuring the Data Server.....	16
Configuring the Logical Schema	18
Create a New Project	19
Reverse-Engineering the Data Models	20
Reverse-Engineering the Oracle Target.....	20
Reverse-Engineering the SAP Source Datastores.....	21
Creating the Data Model	21
Validating the SAP Connection	21
Starting the Reverse-Engineering Process.....	23
Create the Integration Interface.....	28
Create the Interface	28
Define the Source and Target Datastores	29
Define Joins between Sources	30
Create the Mappings	31
Define the Interface Flow	32
Running the Integration Interface.....	34
Running the Interface	34

Review the Interface Execution	34
Review the Resulting Data	35
Conclusion.....	36
Going further with Oracle Data Integrator.....	36
Appendix.....	37
Appendix A - SAP ABAP ERP Required Privileges.....	37
Appendix B - SAP Stand-Alone Connection Test.....	45
Appendix C - Uninstalling ODI SAP Components	47

Introduction

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 Data Integrator - Getting Started with an ETL Project*.

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

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 Staging Area
- Reverse-engineers SAP metadata and proposes a tree browser to pick up only the required metadata

This adapter includes two 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 interfaces. The LKM is in charge of loading source data from a SAP System to an Oracle staging area.

Setting up the Environment

Before You Begin

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 *Oracle Data Integrator Installation Guide* available from the [Oracle Technology Network \(OTN\)](#).

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

- The minimum patch level for Oracle Data Integrator 10gR3 is version 10.1.3.5.1_01 (Metalink Patch [#8610114](#)) or above.
- A JCo version compatible with adapter must be used. The list of supported JCo versions is available in the [Compatibility Matrix](#) available from the 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 the *File Transfer Configurations* section in the *Oracle Data Integrator Knowledge Modules Reference guide*.

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 interface (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.

- SQL*Loader is required on the machine running the agent the executed interfaces using LKM SAP ERP to Oracle (SQLLDR). SQL*Loader is used for loading data extracted from SAP to the Oracle staging area.

Getting the Right Privileges

The SAP Adapter requires privileges to perform set up and execution operations. Please provide your administrators with the list of privileges listed in *Appendix A – SAP ABAP ERP Required Privileges*. These privileges are required for the SAP user that they will provide you to login the SAP System.

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 SNC Connection Properties (optional) SAP Router String (optional):** SAP is enhancing security through SNC and SAP router. It is used when these securities are implemented.
- **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.

Validating the Shared Folder Setup

Validating a shared folder setup applies only, if you plan to transfer data using a Shared Directory. This section can be skipped, if the FTP transfer is used.
The validation of the shared folder setup needs to be performed before any subsequent steps in this guide can be performed. This validation is typically performed by your SAP Basis team.

Step 1: Validating folder access from the SAP application server

1. Start the SAPGUI.
2. Use the ODI SAP user and password to connect to the SAP system and client.
3. Go to transaction AL11.
4. Select the shared folder.
5. Double-click the shared folder to test the directory declaration.

This should report a successful connection. If not, please contact your SAP basis team.
Do not continue until this test passes.

Note: SAP uses the OS user <sid>adm to connect to the directories declared in AL11.

6. Print screen.

Step 2: Validating folder access from the ODI agent machine

1. Log in to the ODI agent machine using the Windows user ID used for executing the ODI agent.
2. Open shared directory path in Windows Explorer.

This should list the content of the shared folder. Please make sure that you do NOT need to enter any credentials.

If this does not work or you had to enter credentials, please contact your windows administrator or the system administrator of the system your shared folder is physically located on.

Do not continue until this test passes.

3. Print screen.

Validating the FTP Setup

Validating an FTP setup applies only, if you plan to transfer data using FTP. This section can be skipped, if you use a Shared Directory for the data transfer.
The validation of the FTP setup needs to be performed before any subsequent steps in this guide can be performed. This validation is typically performed by your SAP Basis team.

Step 1: Validating SAPFTP destination

1. Start the SAPGUI.
2. Use the ODI SAP user and password to connect to the SAP system and client.
3. Go to transaction SM59.
4. Expand TCP/IP connections.
5. Open the SAPFTP destination.
6. Click **Test connection**.

This should report a successful connection. If not, please contact your SAP basis team.
Do not continue until this test passes.

7. Print screen.

Step 2: Testing FTP connection

1. Go to transaction SE38.
2. View the function module RSFTP002.
3. Hit F8 to run the ABAP program.
4. Enter the FTP userID and password.
5. Enter the FTP server host name or IP address.
6. Enter `cd /` or `cd <ODI target directory>`.
7. In the RFC_DESTINATION field, enter `SAPFIPA`.
8. Hit F8 to run the test.

This should report a successful connection. The message should be similar to the following:

```
250 CWD successful.
```

If this test is not successful, please contact your SAP basis team. Do not continue until this test passes.

9. Print screen.

Validating SAP Privileges

This section describes how to test some of the key SAP privileges. Proceed with the subsequent steps in this guide only after successful validation of these tests. This validation is typically performed by your SAP Basis team.

Perform the following steps to validate whether a SAP user has appropriate dev rights and owns a dev license key:

1. Start SAPGUI.
2. Use the ODI SAP user and password to connect to the SAP system and client.
3. Go to transaction SE38.
4. Enter any sample program name like `ZSAP_TEST` in the program name field.
5. Click **Create**.
6. Perform similar tests for the transaction SE37 and SE11.

If a transaction allows the creation of a program without asking for any key or other authorization message, then the SAP user has validated that it has the appropriate dev rights and license key. Otherwise your SAP basis team needs to register the SAP user in service.sap.com to get the license key and a Basis person can help him with dev rights.

Validating SAP Transport Layer Name

As the SAP connector creates SAP objects, such as, for example, function modules, into the SAP development system, these changes need to be transported into QA and production systems once the development is done. The SAP's change and transport system uses the *SAP Transport Layer Name* to identify the route a change has to take. A transport layer is assigned to each development class and thus to all objects in that class. The transport layer determines:

- In which SAP System developments or changes to the repository objects are made
- If objects are transported to other systems within the group when development work has been completed

A consolidation route is created from the development system to the quality assurance system through the `transport layer Z<SID>`. It then becomes the standard transport layer for customer development and customizing.

A consolidation route is created from the development system to the quality assurance system through the `transport layer SAP` for the transport of SAP Standard objects.

It is important to specify the correct transport layer name before running the RKM SAP ERP for the first time. Perform the following steps to identify the list of defined transport layers in your SAP landscape:

1. Log on in client 000 in the SAP System serving as the transport domain controller via transaction STMS.
2. Select **Overview > Transport Routes**. The Display Transport Routes dialog is displayed.
3. Select **Goto > Graphical Editor**.
4. To switch the mode, select **Configuration > Display <-> Change**.
5. Position the cursor on the SAP System.
6. Select **Edit > System > Change**. The Change System Attributes dialog is displayed.
7. Select the StandardTransport Layer tab.
8. Change the transport layer of the SAP System.
9. The result is the list of the different transport Layers.

By default, the RKM option `SAP_TRANSPORT_LAYER_NAME` is set to `SAP`. Ask your SAP basis admin which transport layer you should use. This transport layer name must be set on the `SAP_TRANSPORT_LAYER_NAME` RKM option. A wrong or invalid transport layer name will cause serious delays during the installation process.

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 *Oracle Fusion Middleware Installation Guide for 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 integration interfaces. You can download for example an Oracle XE database from the [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 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 [Compatibility Matrix](#) available at Oracle Technology Network. Note that a minimum version of JCo 3.0.2 is required.
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 (or respective binary) into the oracledi/drivers directory.
5. Check the JCo installation. This will be also checked later in this guide.

Note: Changing the JCo library installed in the {windows-dir}\system32 directory of a machine running other SAP tools or components may cause issues with these components. Please check with your machine administrator before performing this change.

Setting up an FTP Server

The SAP adapter extracts SAP data and uploads it to an FTP server. Configure an FTP server or use an existing FTP server. You must create a user in this FTP server and a directory into which this user will be able to upload files.

Configuring Oracle Data Integrator

Set up the Topology

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 [Oracle Fusion Middleware Installation Guide for Oracle Data Integrator](#) on OTN 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 server into which the extraction file will be pushed from SAP and picked up for SQL*Loader. Set the parameters for this data server as follows:
 - **Host (Data Server):** FTP server IP host name or IP address.
 - **User:** Username to log into FTP server.
 - **Password:** Password for the user.
6. In this File data server create a Physical Schema representing the folder in the FTP host where the extraction file will be pushed. Specify the Data and Work Schemas as follows:
 - **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) 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`.

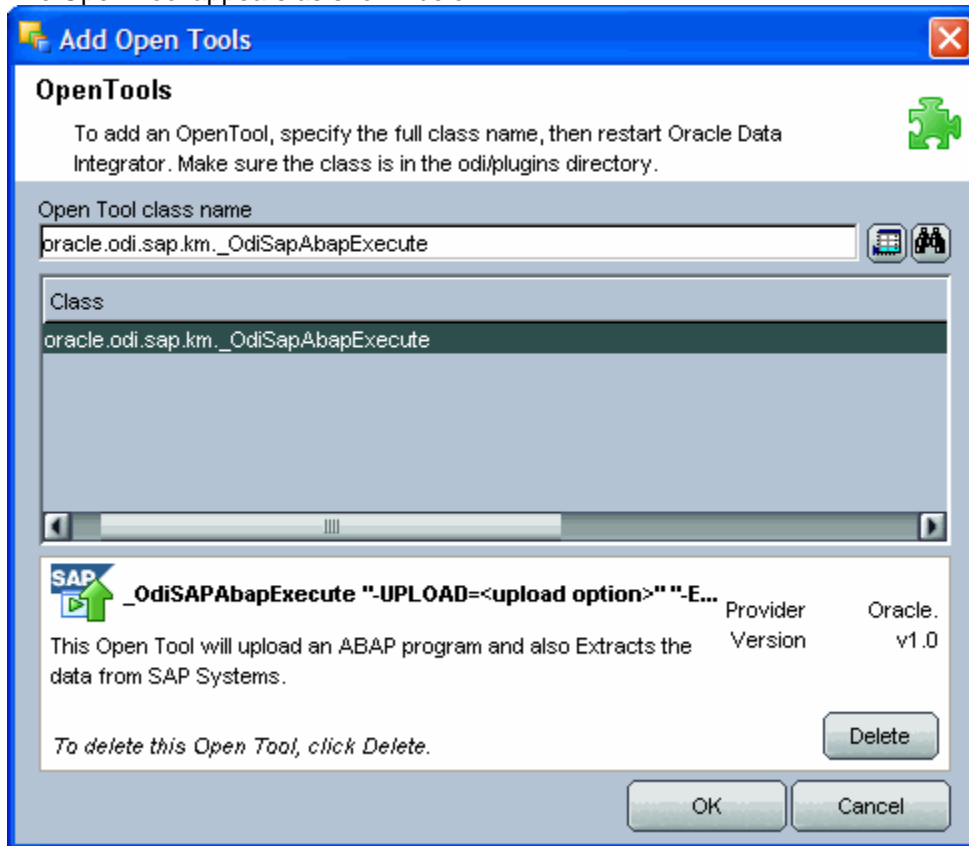
Refer to the *File Transfer Configurations* section in the *SAP ABAP* chapter of the *Oracle Fusion Middleware Application Adapters Guide for Oracle Data Integrator*.
7. If the corresponding Logical Schema called `File Server for SAP ABAP` does not exist yet, create it. This Logical Schema name is fixed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

Add the Open Tool

The complete process of installing and adding an Open Tool to ODI is described in *the Oracle Data Integrator Tools Reference in the Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*. 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._OdiSapAbapExecute`
4. Click **Add Open Tool**.

5. The Open Tool appears as shown below.



6. Click **OK**.

Configuring the Topology

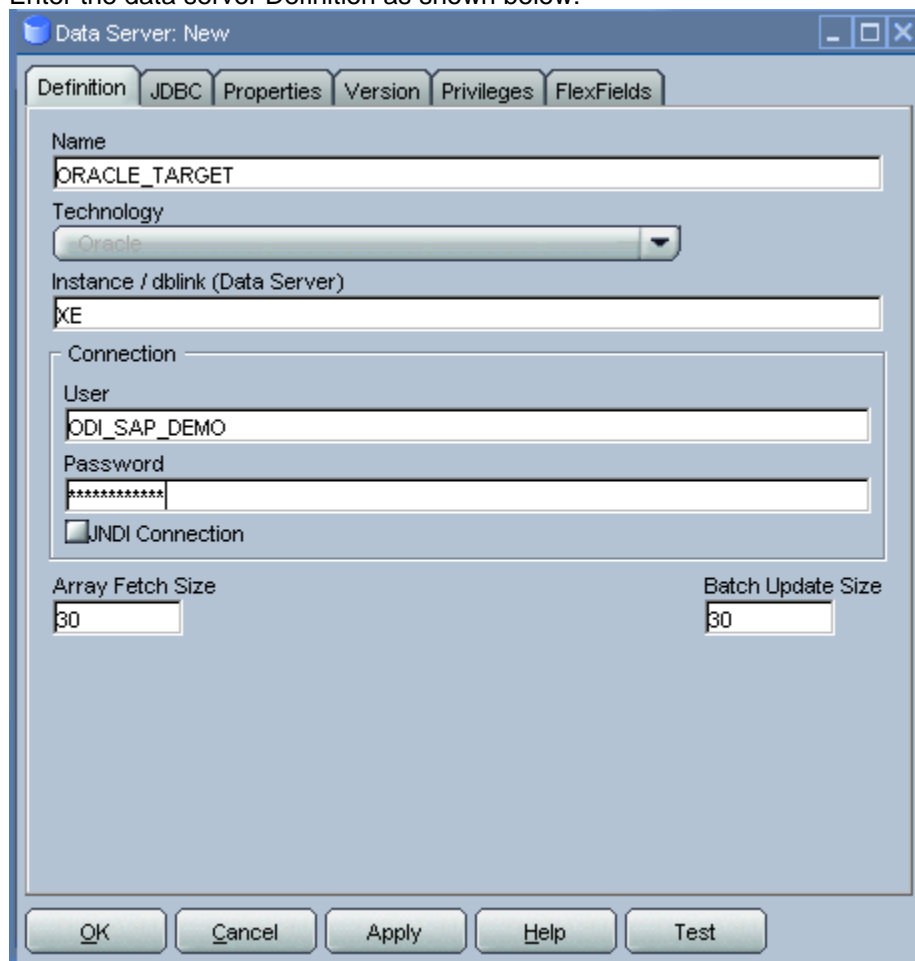
This project will integrate the following source and target data:

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`.

Configuring the Oracle Data Server

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 **Insert Data Server**.
4. Enter the data server Definition as shown below:

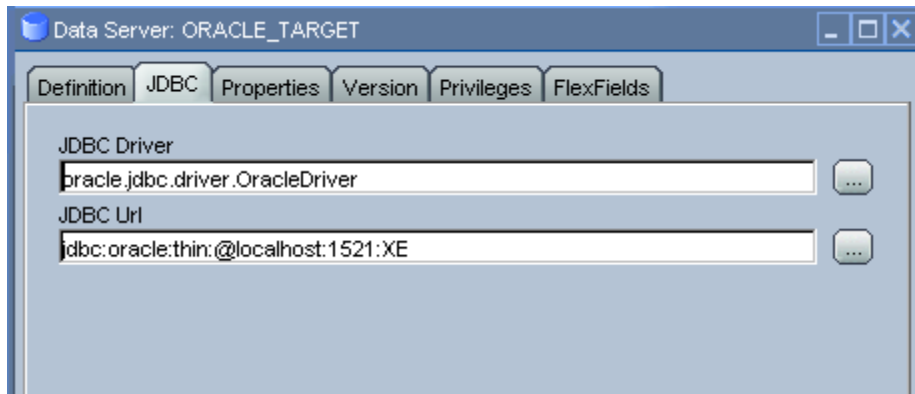


The screenshot shows the 'Data Server: New' dialog box with the 'Definition' tab selected. The fields are filled as follows:

- Name:** ORACLE_TARGET
- Technology:** Oracle (selected in a dropdown menu)
- Instance / dblink (Data Server):** XE
- Connection:**
 - User:** ODI_SAP_DEMO
 - Password:** (masked with asterisks)
 - JNDI Connection:** (unchecked checkbox)
- Array Fetch Size:** 30
- Batch Update Size:** 30

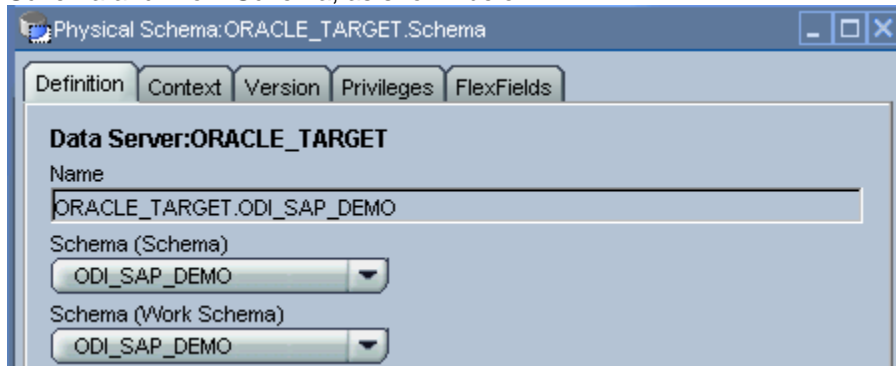
At the bottom, there are buttons for OK, Cancel, Apply, Help, and Test.

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



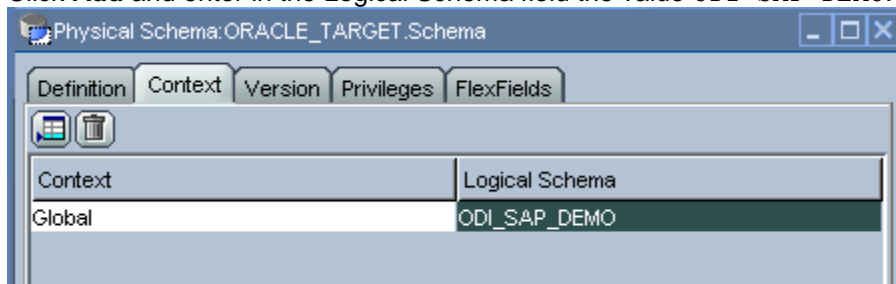
Make sure to change the URL to match your Oracle instance configuration.

6. Click **Test** to test the connection to this data server.
7. Click **OK** to save your data server.
8. In the Physical Schema windows that appears, select the `ODI_SAP_DEMO` for both Schema and Work Schema, as shown below:



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.

9. Go to the Context tab.
10. Click **Add** and enter in the Logical Schema field the value `ODI_SAP_DEMO`.



11. Click **OK** to save.

Configuring the SAP Source Server

Configuring the Data Server

1. In the Physical Architecture tree view, expand the Technologies node, select the `SAP ABAP` technology.

2. Right-click and select **Insert 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.

Data Server: New

Definition | JDBC | Properties | Version | Privileges | FlexFields

Name: SAP_ERP

Technology: SAP ABAP

Host (Data Server): saphostname

Connection:

User: saplogon

Password: *****

☐ JNDI Connection

4. Set the Flexfield values for this data server in the Flexfields tab.

Data Server: New

Definition | JDBC | Properties | Version | Privileges | FlexFields

Name	Defa...	Value
SAP SNC Connection Properties	<input checked="" type="checkbox"/>	
SAP Client Number	<input checked="" type="checkbox"/>	<Please specify SAP Client Number>
SAP Language	<input checked="" type="checkbox"/>	EN
SAP System Number	<input checked="" type="checkbox"/>	<Please specify SAP System No>
SAP Allow ABAP Upload	<input checked="" type="checkbox"/>	1
SAP System ID	<input checked="" type="checkbox"/>	<Please specify SAP System ID>
SAP Router String	<input checked="" type="checkbox"/>	

- **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 SNC Connection Properties:** SNC Connection Properties. This parameter is optional and can be left empty.
- **SAP Router String:** Router String. This parameter is optional and can be left empty.
- **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`.
- **SAP ABAP Version:** Enter the SAP ABAP version as follows:
For SAP 4.6C enter `46C`
For SAP ECC 6.0 enter `700`
- **SAP BW Version & SAP ABAP Version:** unused in SAP ERP Connector.

5. Click **OK**.

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

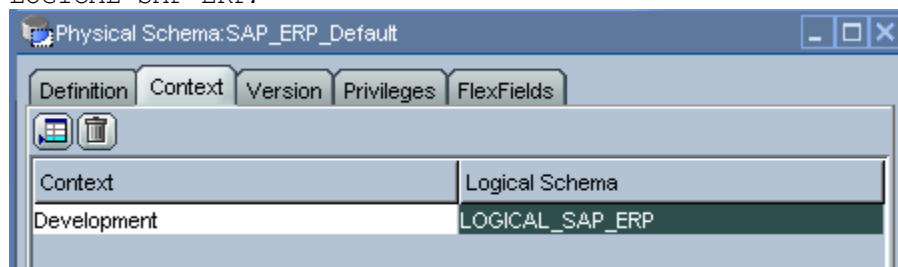
Except for the data server name, all the parameters that you provide while defining the SAP Data server should be provided by the SAP Administrators. See *Getting the Right Privileges* section for more information.

The SAP Adapter requires privileges to perform set up and execution operations. Please provide your administrators with the list of privileges listed in *Appendix A - SAP ABAP ERP Required Privileges*. These privileges are required for the SAP user that they will provide you to login the SAP System.

See *Gathering SAP Connection Information* in the *Before You Begin* section for more information about these parameters.

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. Click **OK**.

Create a New Project

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.

To create a new Project for SAP:

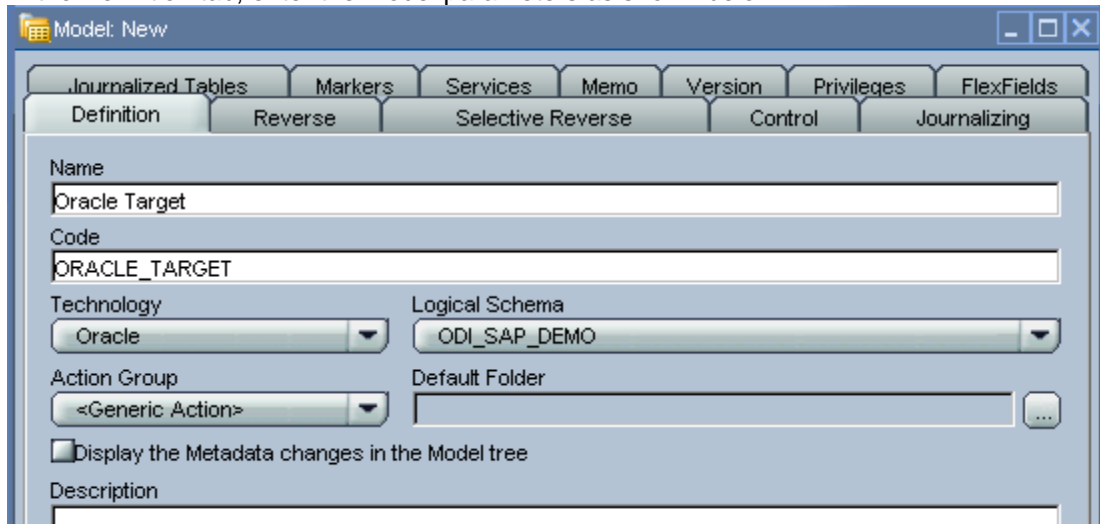
1. Connect to Designer.
2. In the Projects tree view click **Insert Project**.
3. In the Definition tab, enter `SAP Demo` for the Project name.
4. Click **OK** to save your project.
5. Expand the `SAP Demo Project`, select the `First Folder` node.
6. Rename this folder to `SDE_SAP_GeoDimension`.
7. Right-click the `SAP Demo Project` in the Projects tree view.
8. Right-click and select **Import 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)
 - RKM SAP ERP
 - RKM SAP ERP Connection Test
11. Click **OK** to perform the import.

Reverse-Engineering the Data Models

Reverse-Engineering the Oracle Target

To reverse-engineer the Oracle data model:

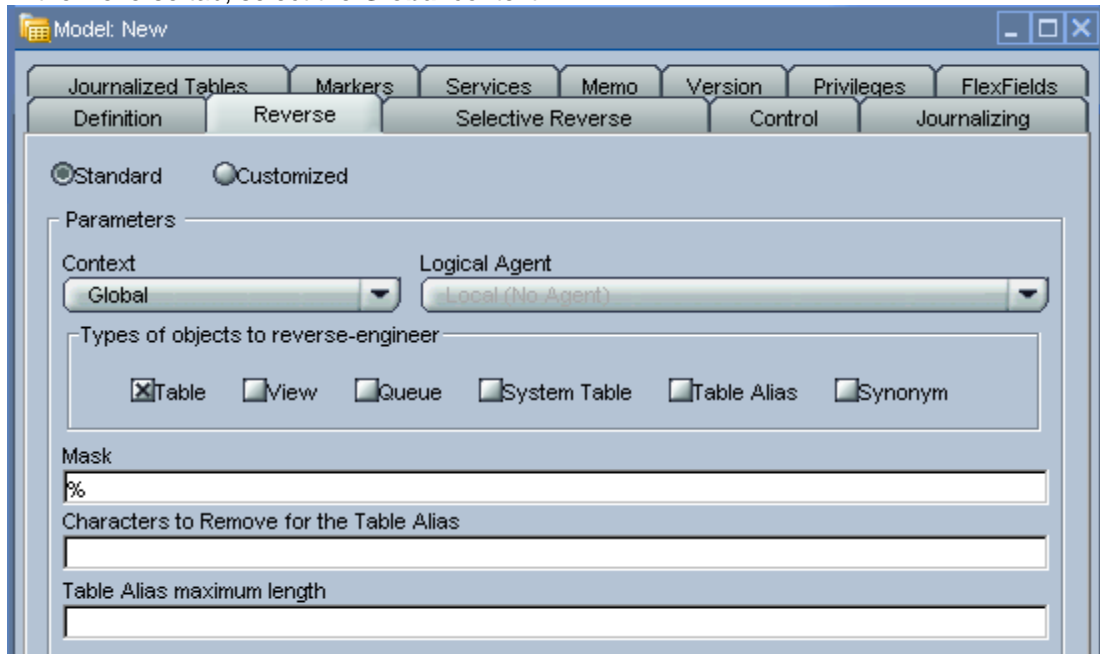
1. Connect to Designer.
2. In the Models tree view, click **Insert Model**.
3. In the Definition tab, enter the model parameters as shown below:



The screenshot shows the 'Model: New' dialog box with the 'Definition' tab selected. The fields are filled as follows:

- Name: Oracle Target
- Code: ORACLE_TARGET
- Technology: Oracle
- Logical Schema: ODI_SAP_DEMO
- Action Group: <Generic Action>
- Default Folder: (empty)
- ☐ Display the Metadata changes in the Model tree
- Description: (empty)

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

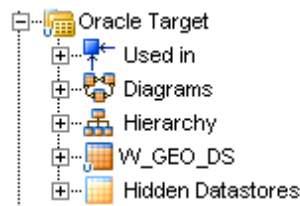


The screenshot shows the 'Model: New' dialog box with the 'Reverse' tab selected. The settings are as follows:

- ☒ Standard ☐ Customized
- Context: Global
- Logical Agent: Local (No Agent)
- Types of objects to reverse-engineer:
 - ☒ Table
 - ☐ View
 - ☐ Queue
 - ☐ System Table
 - ☐ Table Alias
 - ☐ Synonym
- Mask: %
- Characters to Remove for the Table Alias: (empty)
- Table Alias maximum length: (empty)

5. Click **Reverse**, then **OK** to 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. Create the Data Model
2. Validate the SAP Connection
3. Start the Reverse-Engineering Process

Creating the Data Model

1. In the Models tree view, click **Insert 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).

The screenshot shows the 'Model: New' dialog box. It has several tabs: 'Definition', 'Reverse', 'Selective Reverse', 'Control', and 'Journalizing'. The 'Definition' tab is selected. The dialog contains the following fields and controls:

- Name:** SAP ERP Source
- Code:** SAP_ERP_SOURCE
- Technology:** SAP ABAP (dropdown menu)
- Logical Schema:** LOGICAL_SAP_ERP (dropdown menu)
- Action Group:** <Generic Action> (dropdown menu)
- Default Folder:** (empty text field with a browse button)
- ☐ Display the Metadata changes in the Model tree
- Description:** (empty text area)

3. In the Reverse 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** 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 the Pre-requisites section 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.

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 Appendix B - SAP Stand-Alone Connection Test.

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 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. For first time RKM use only: Set `SAP_TRANSPORT_LAYER_NAME` to the name of the transport layer your SAP basis admin has provided you with.

Caution: Pay particular attention to provide the correct value for the `SAP_TRANSPORT_LAYER_NAME` option. If an invalid value is used during the first time installation, a significant delay in the connector installation may be caused.

- f. For first time RKM use only: Set `UPLOAD_ABAP_CODE` to `Yes`
After the first execution this value must be set back to No.
 - g. Set other parameters according to your SAP configuration. See the *Oracle Fusion Middleware Application Adapters Guide for Oracle Data Integrator* for more information on the RKM options.

Model: SAP ERP Source

Journalized Tables Markers Services Memo Version Privileges FlexFields

Definition Reverse Selective Reverse Control Journalizing

☐ Standard ☒ Customized

Parameters

Context: Global Logical Agent: Local (No Agent)

Types of objects to reverse-engineer:

☒ Table ☐ View ☐ Queue ☐ System Table ☐ Table Alias ☐ Synonym

Mask: %

Characters to Remove for the Table Alias:

Select your KM: RKM SAP ERP.SAP Demo

Option	Value
USE_GUI	<Default>:Yes
SAP_MODULE_NAME	<Default>:%
SAP_APP_COMP_NAME	<Default>:%
SAP_PACKAGE_NAME	<Default>:%
SAP_TABLE_DESC	<Default>:%
SAP_TABLES_NAME	<Default>:
SAP_CONNECTION_POOL	<Default>:SAP_POOL_NAME_ECC6
SAP_CONNECTION_POOL_SIZE	<Default>:20
GET_FOREIGN_KEYS	<Default>:Yes
GET_INDEXES	<Default>:Yes
GET_PRIMARY_KEYS	<Default>:Yes
LOG_FILE_NAME	Flag to get indexes java.lang.System.getProperty("jav...
UPLOAD_ABAP_CODE	<Default>:Yes
SAP_FUNCTION_GROUP	<Default>:SAP_ERP_FUN_GRP

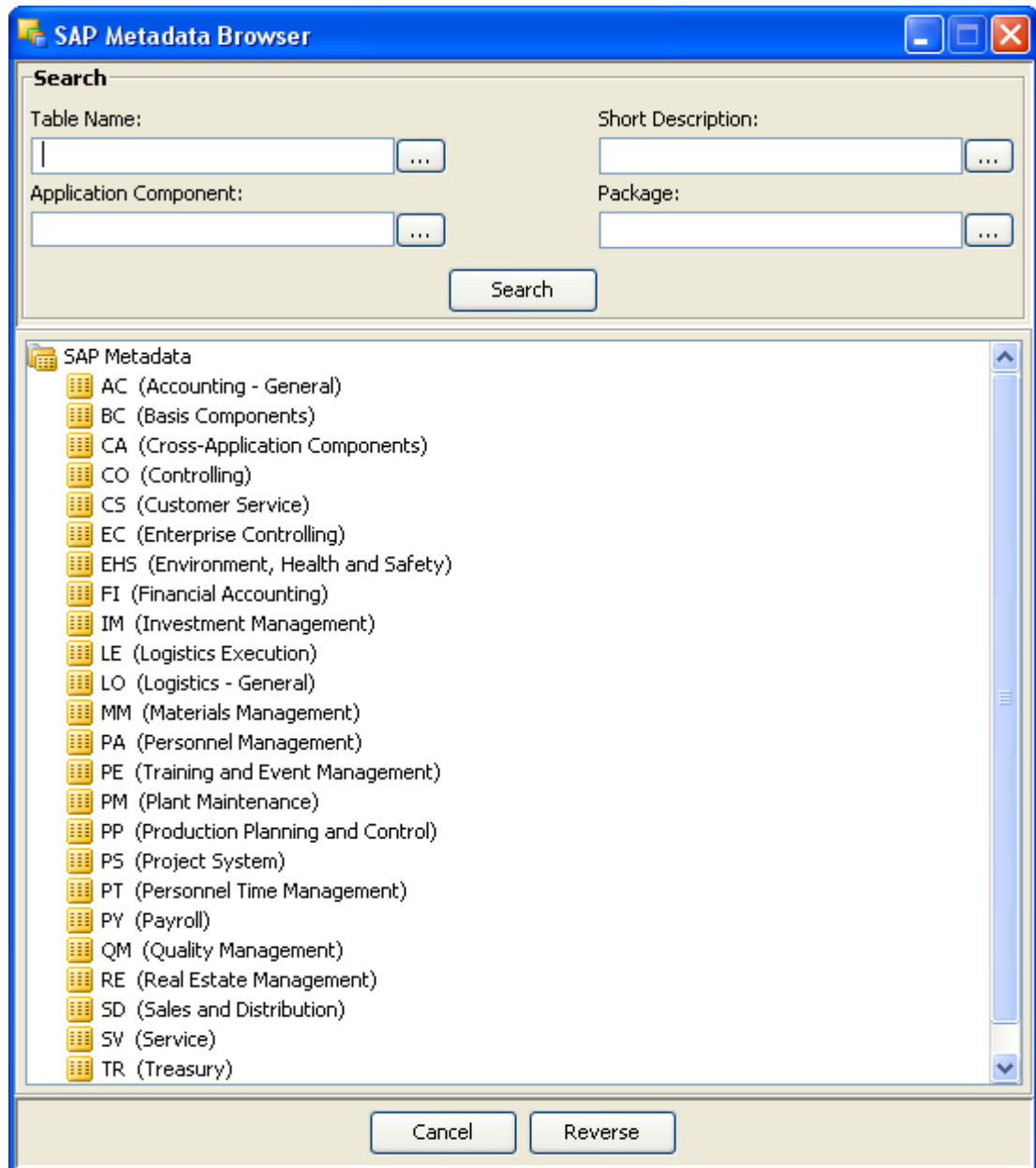
3. Click **Apply** to save your changes.
4. Click **Reverse** to start the reverse-engineering process.
5. Click **OK**.
6. The Sessions Started Dialog is displayed.
7. Click **OK**.
8. Set UPLOAD_ABAP_CODE back to No.
9. For first time RKM use only: Validate in Operator that the session is now in status 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 for details and read the log files `<System Temp Dir>/sap_rkm_erp_<ODI Session Number>.log` and `<System Temp Dir>/sap_rkm_erp_<ODI Session Number>.log`.

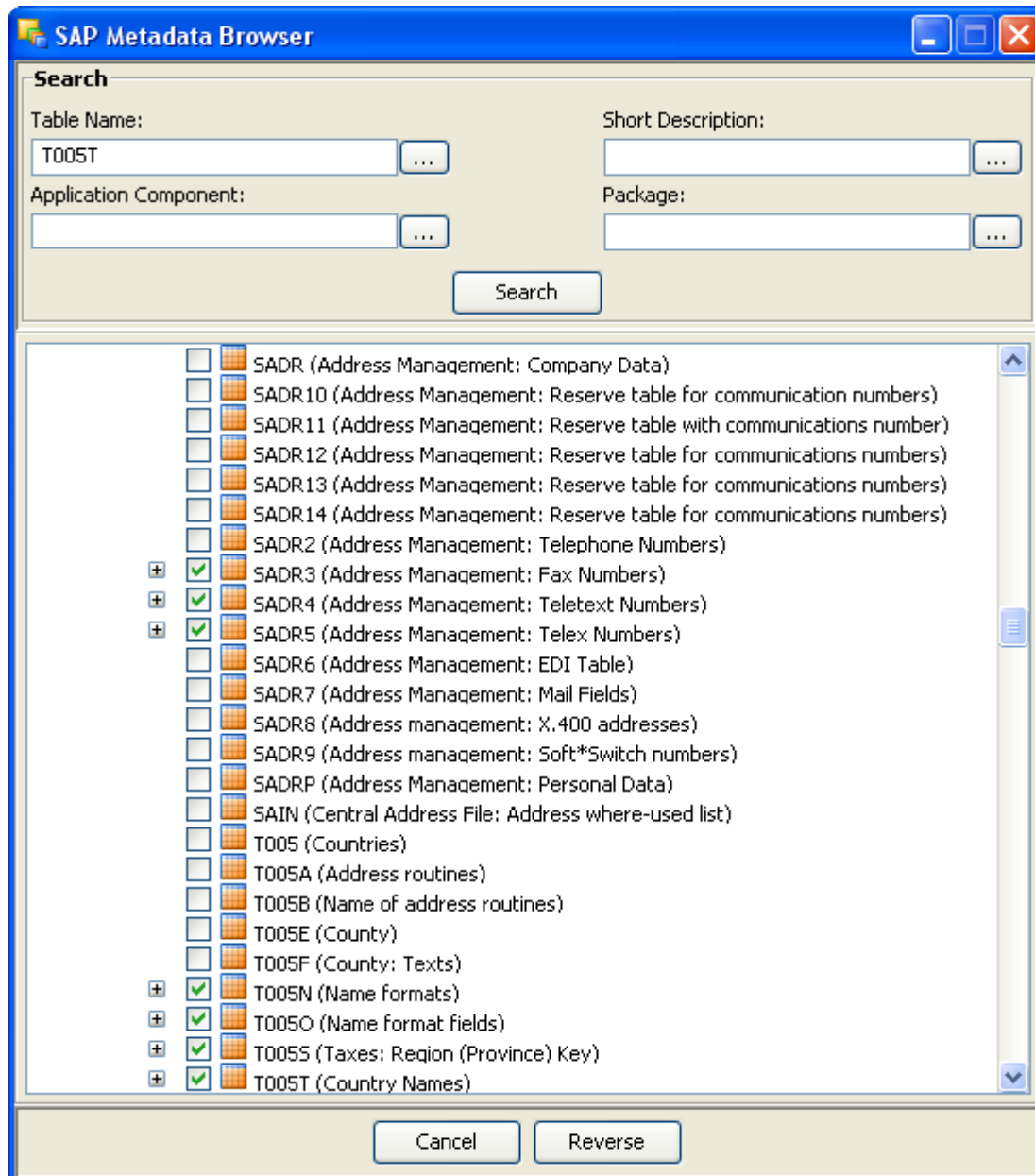
Refer to the *Log Files* section in the *SAP ABAP ERP* chapter of the *Oracle Fusion Middleware Application Adapters Guide for Oracle Data Integrator* for more details.



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 *Oracle Data Integrator Knowledge Modules Reference Guide* 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 option is selected (this option is typically activated only once). The ABAP programs are generated in the SAP function group given in the SAP_FUNCTION_GROUP RKM option.

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.

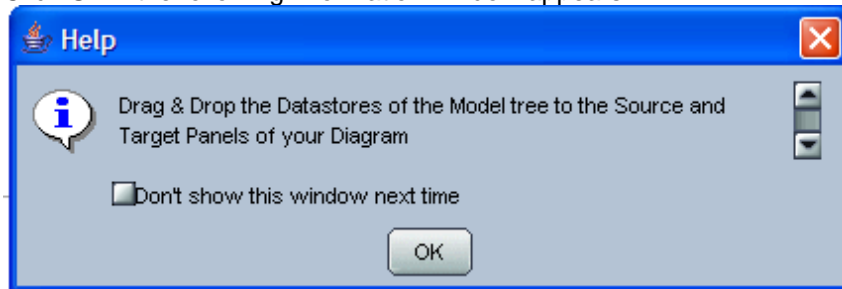
In addition to the table definitions organized and enriched with user-friendly information, the keys and indexes definitions are also retrieved. The model appearing in Oracle Data Integrator shows all the information required for performing integration tasks on the SAP ERP Model.

Create the Integration Interface

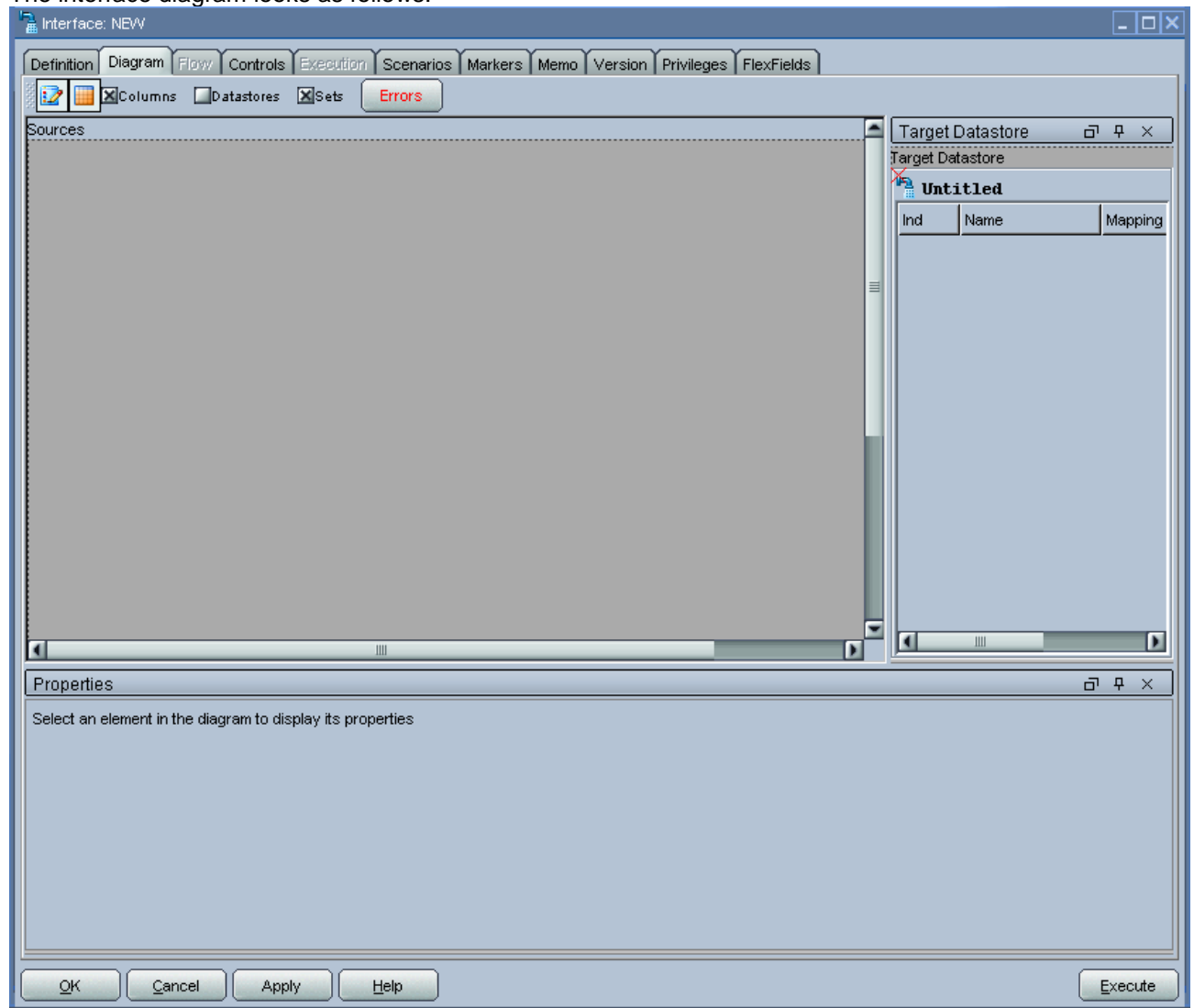
Now that the source and target data models are created, it is possible to create an interface to integrate data from the T005T and T005U SAP tables to the WS_GEO_DS Oracle table.

Create the Interface

1. Open Designer.
2. In the Projects tree view, expand the SAP Demo Project.
3. Expand the SDE_SAP_GeoDimension folder.
4. Select the Interfaces node.
5. Right-click and select **Insert Interface**.
6. In the Interface Definition tab, enter the interface name:
SDE_SAP_GeoDimension.W_GEO_DS.
7. Select the Diagram tab.
8. Click **OK** if the following information window appears.



The interface diagram looks as follows:

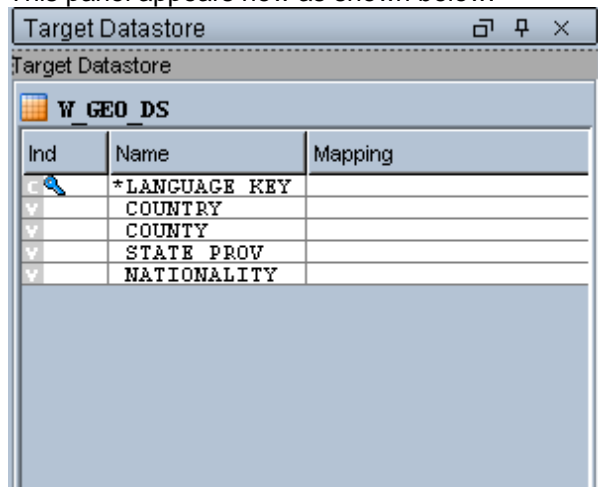


Define the Source and Target Datastores

To define the source and target datastores:

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

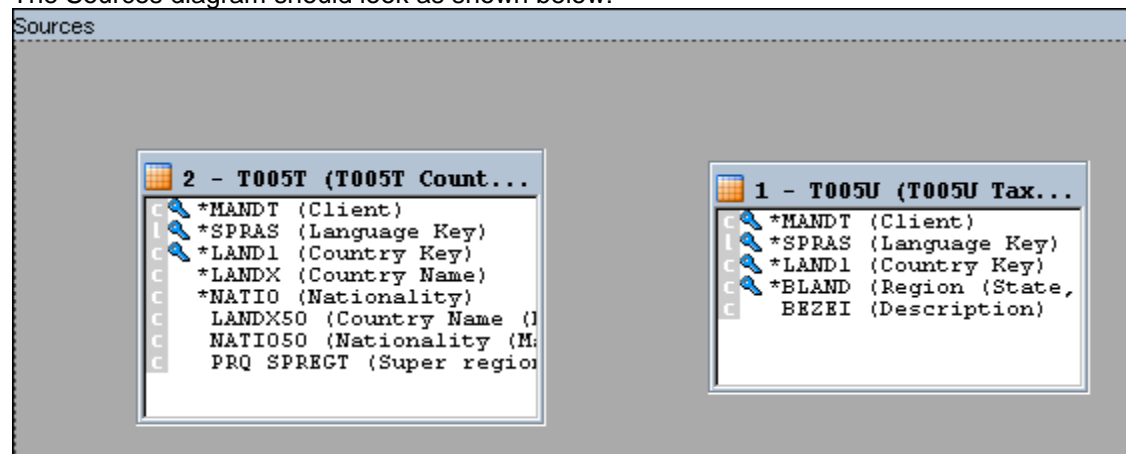
2. Drag this datastore into the Target Datastore panel (right area of the Diagram tab). This panel appears now as shown below:



Ind	Name	Mapping
	*LANGUAGE KEY	
	COUNTRY	
	COUNTRY	
	STATE PROV	
	NATIONALITY	

3. In the Models tree view, select the T005T Country Names datastore from the SAP ERP Source model.
4. Drag this datastore into the Sources diagram panel (left area of the diagram tab).
5. If a popup window prompting “Do you want to perform Automatic Mapping?” appears, click **No**.
6. Perform the same operation for the T005U Taxes: Region Key datastore from the SAP ERP Source model.

The Sources diagram should look as shown below:

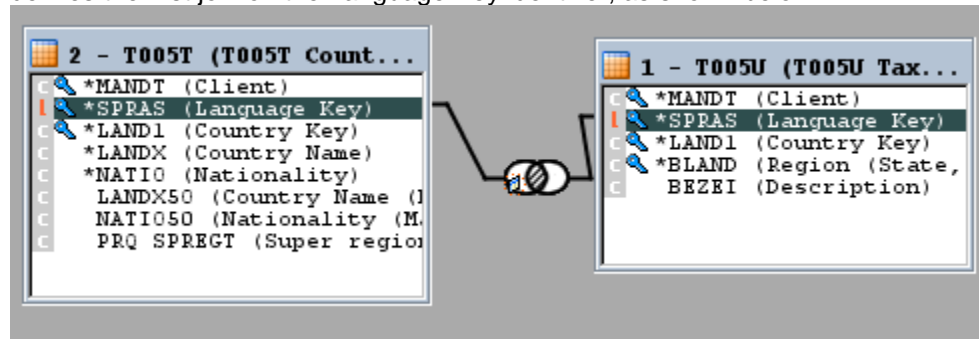


Define Joins between Sources

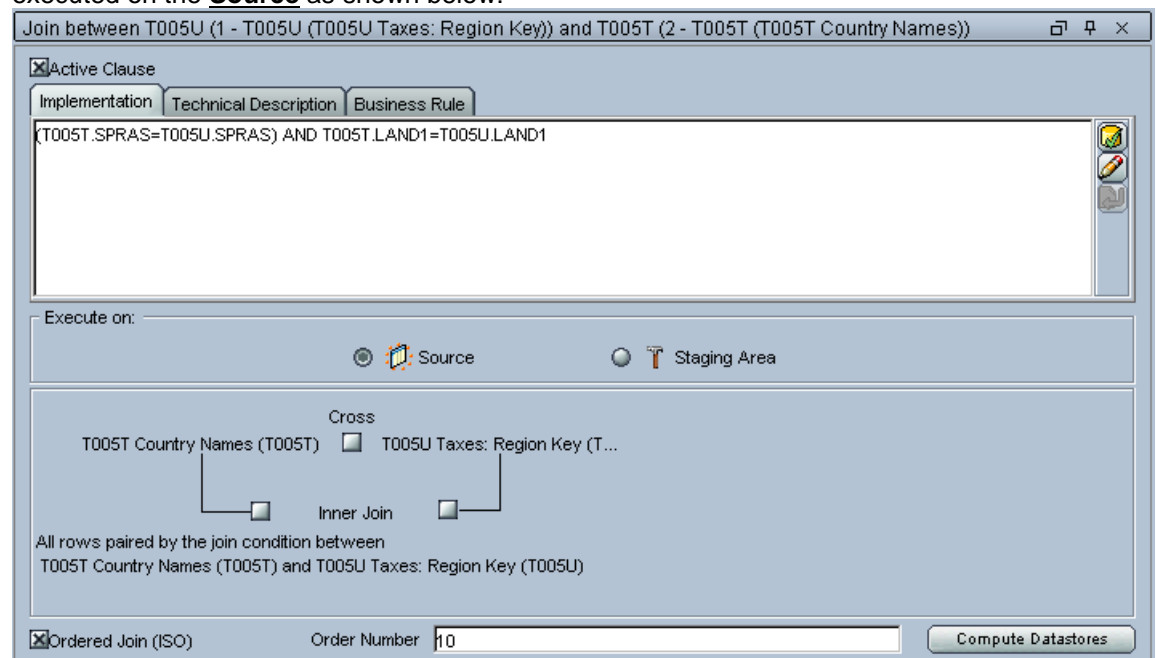
To create joins between the source datastores of an interface:

1. In the Sources diagram drag the SPRAS column from the T005T Country Name datastore onto the SPRAS column in the T005U Taxes: Region Key table. This

defines the first join on the Language Key identifier, as shown below:



2. Drag the LAND1 column from the T005T Country Name table onto the LAND1 column in the T005U Taxes: Region Key table. The join is extended with the new columns. In the Properties panel, you can see the join clause:
(T005T.SPPAS=T005U.SPPAS) AND T005T.LAND1=T005U.LAND1
3. In the Properties panel, select **Ordered join** and make sure that this join is executed on the **Source** as shown below:



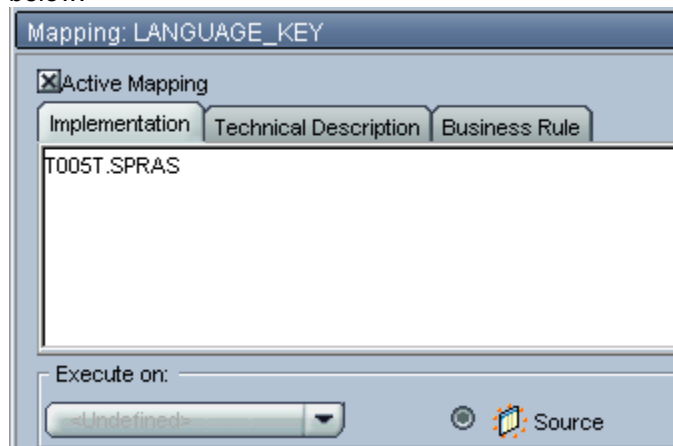
4. Make sure that table number of T005U is smaller than that of T005T. In 1:n relationships the parent table has to be the first in a join.

Create the Mappings

To create the mappings of the target datastore:

1. Select the LANGUAGE KEY column from the target datastore.
2. Drag the SPPAS column from the T005T Country Names source table into the Implementation field of the Properties panel.

3. Make sure that the **Execution On** is set to **Source**. The mapping should look as shown below:

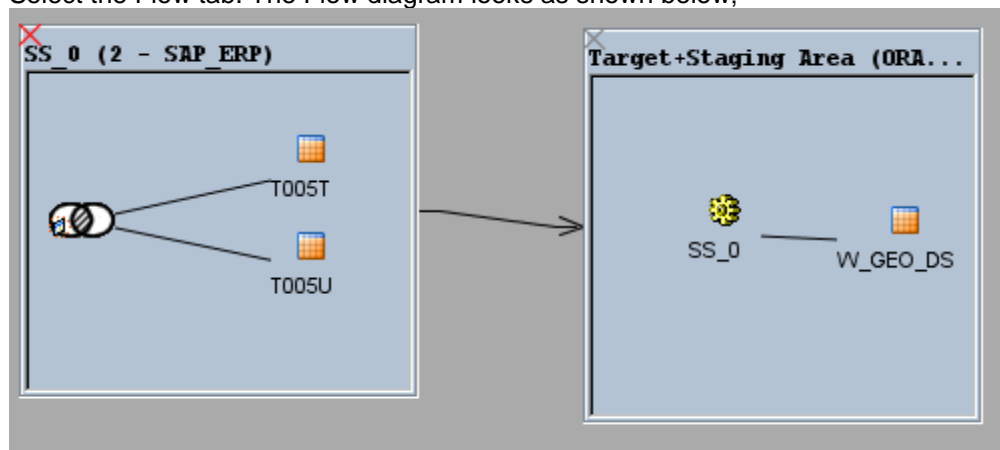


4. Repeat this operation to perform the following simple mappings:
 - W_GEO_DS.COUNTRY = T005T.LAND1
 - W_GEO_DS.COUNTY = T005T.LANDX
 - W_GEO_DS.STATE_PROV = T005U.BLAND
 - W_GEO_DS.NATIONALITY = T005T.NATIO

Define the Interface Flow

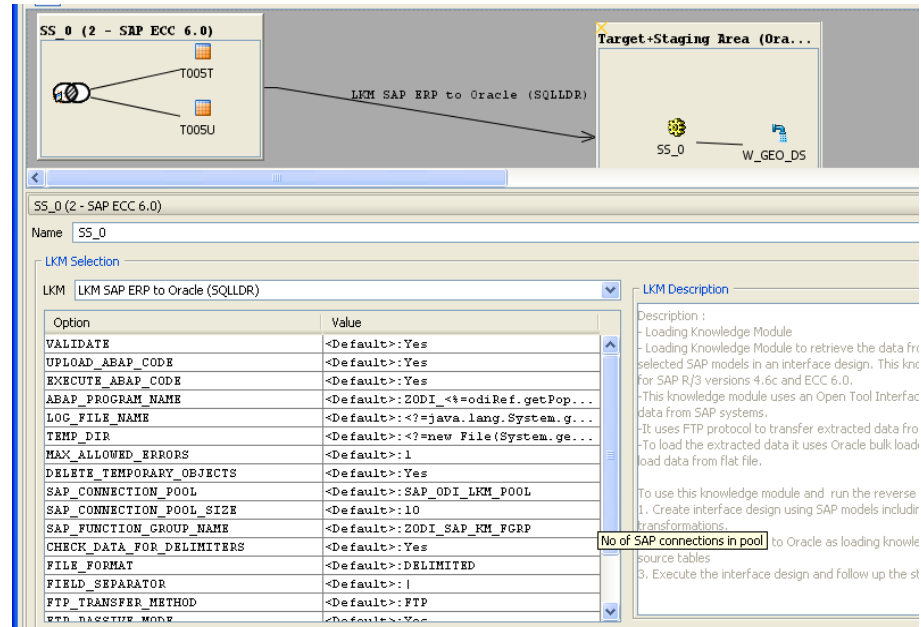
To define the interface flow:

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



2. Select the source set **SS_0 (SAP_ERP)** that contains both the **T005T** and **T005U** tables. This source set represents the source dataset made up of two joined **SAP ERP** tables.
3. In the LKM Selection box, select **LKM SAP ERP to Oracle (SQLLDR)**.
4. Select the **Target + Staging Area** box. This represents the target Oracle database, which is also used as the staging area for this interface.
5. In the LKM Selection, select **LKM SAP ERP to Oracle (SQLLDR)**.
6. In the IKM Selection, select **IKM Oracle Incremental Update**.
7. Set the IKM options as follows:

- Select **No** for the *FLOW_CONTROL* option.
- 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**.
- Leave other options as is. The flow should look as shown below:



- Click **OK** to save and close your interface.

Running the Integration Interface

Running the Interface

To run the integration interface:

1. In the Projects tree view, expand the `SAP Demo Project`
2. Expand the `SDE_SAP_GeoDimension` folder.
3. Expand the Interfaces node.
4. Select the `SDE_SAP_GeoDimension.W_GEO_DS` interface.
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 interface?

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

1. 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.
2. 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_NAME` 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 LKM option.

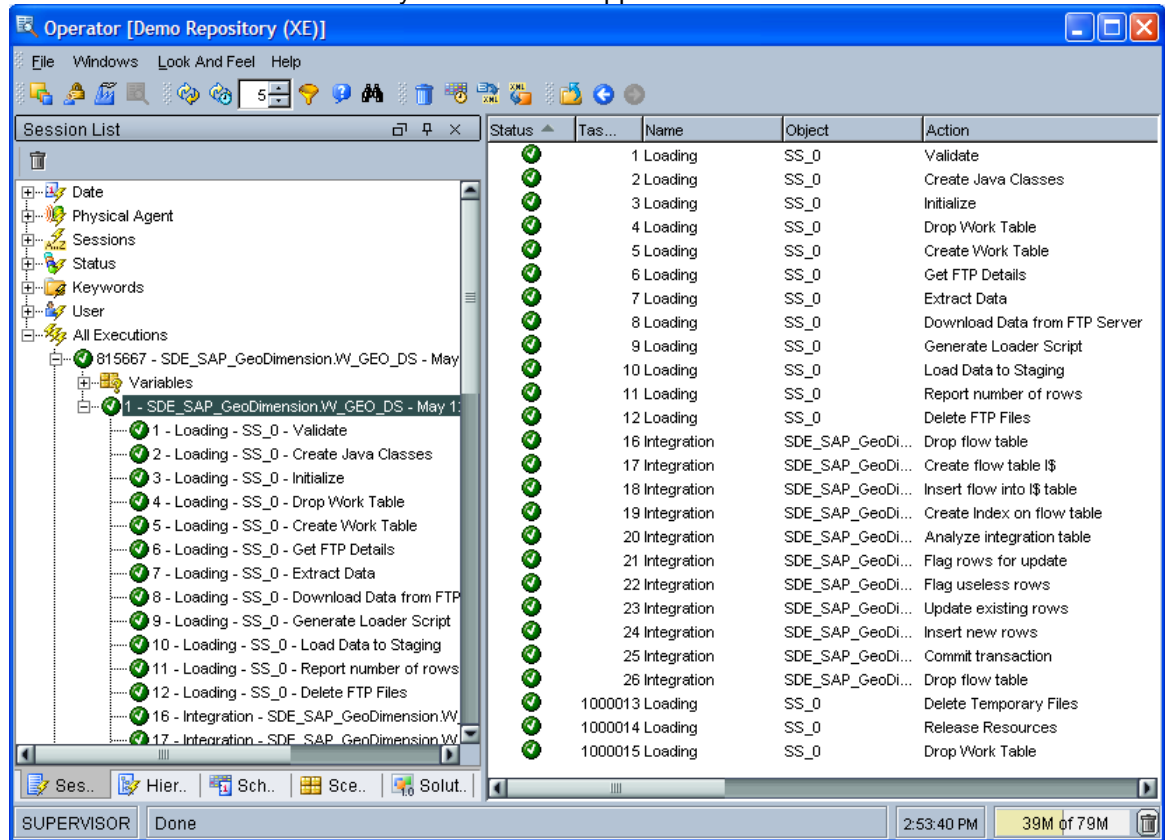
Finally, when the agent accesses the extraction file, it uses `SQL*Loader` to load this file into the Oracle staging area. The rest of the integration process takes place within the Oracle engine.

Review the Interface Execution

To review the interface 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 and appears as shown below.



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.

Conclusion

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 an interface to load the data from several joined SAP ERP tables into the Oracle database.

Refer to the *Oracle Data Integrator Knowledge Modules Reference Guide* for more information on the SAP ERP KM features and options.

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

- <System Temp Dir>/ODI_<Interface Id>_<SrcSet>.genlog
- <System Temp Dir>SAPAbapExecuteOpenTool_<Interface Id>.log
- <System Temp Dir or local FTP dir>/ ZODI_<Interface Id>_<SrcSet>_<Context>.log
- <System Temp Dir or local FTP dir>/ ZODI_<Interface Id>_<SrcSet>_<Context>.out
- <System Temp Dir or local FTP dir>/ ZODI_<Interface Id>_<SrcSet>_<Context>.err

Refer to the *Log Files* section in the *SAP ABAP* chapter of the *Oracle Data Integrator Knowledge Modules Reference Guide* for more details.

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](#).

Appendix

Appendix A - SAP ABAP ERP Required Privileges

A SAP dialog user is required for connecting to the SAP system, for the RKM installation and for the LKM execution during development. A SAP RFC user is NOT sufficient.

During development, this user must have a developer license key. License keys can be retrieved from <http://service.sap.com>. Any execution attempts without this developer license key will lead to failure and may cause the need for clean-up operations.

For RKM and LKM execution in production, a SAP RFC user is sufficient.

The following tables list the privileges required for using SAP ABAP Knowledge Modules.

Authorizations Required for RKM SAP ERP Setup

The following authorizations are required for setting up the configuration for the **RKM SAP ERP** and install the required RFC programs to retrieve the metadata about SAP tables.

Object	Field Name	Value
S_ADMI_FCD	S_ADMI_FCD	*
S_ADMI_FCD	S_ADMI_FCD	NADM
S_ADMI_FCD	S_ADMI_FCD	PADM
S_CTS_ADMI	CTS_ADMFCT	EPS1
S_CTS_ADMI	CTS_ADMFCT	EPS2
S_CTS_ADMI	CTS_ADMFCT	PROJ
S_C_FUNCT	ACTVT	16
S_C_FUNCT	CFUNCNAME	*
S_C_FUNCT	PROGRAM	*
S_CTS_ADMI	CTS_ADMFCT	TABL
S_TCODE	TCD	SU53
S_TCODE	TCD	SE38
S_TCODE	TCD	SE80
S_DATASET	ACTVT	6
S_DATASET	ACTVT	33
S_DATASET	ACTVT	34
S_DATASET	FILENAME	*

S_TCODE	TCD	SE91
S_TCODE	TCD	ST22
S_DATASET	PROGRAM	*
S_RFC	RFC_TYPE	*
S_RFC	RFC_NAME	*
S_RFC	ACTVT	16
S_DEVELOP	ACTVT	*
S_DEVELOP	DEVCLASS	*
S_DEVELOP	OBJNAME	*
S_DEVELOP	OBJTYPE	*
S_DEVELOP	P_GROUP	*
S_DOKU_AUT	DOKU_ACT	RAW_VERS
S_DOKU_AUT	DOKU_DEVCL	TEST
S_DOKU_AUT	DOKU_MODE	MAINTAIN
S_RFC_ADM	ACTVT	3
S_RFC_ADM	ICF_VALUE	*
S_RFC_ADM	RFCDEST	*
S_RFC_ADM	RFCTYPE	*
S_RZL_ADM	ACTVT	*
S_TABU_DIS	ACTVT	3
S_TABU_DIS	DICBERCLS	*
S_TCODE	TCD	■ AL11
S_TCODE	TCD	SE10
S_TCODE	TCD	SE11
S_TCODE	TCD	SE16
S_TCODE	TCD	SE37
S_TCODE	TCD	SM58
S_TCODE	TCD	SM59
S_TRANSPRT	ACTVT	1
S_TRANSPRT	ACTVT	2
S_TRANSPRT	ACTVT	3

S_TRANSPRT	ACTVT	5
S_TRANSPRT	ACTVT	6
S_TRANSPRT	ACTVT	23
S_TRANSPRT	ACTVT	43
S_TRANSPRT	ACTVT	65
S_TRANSPRT	ACTVT	78
S_TRANSPRT	ACTVT	90
S_TRANSPRT	TTYPE	CUST
S_TRANSPRT	TTYPE	DTRA
S_TRANSPRT	TTYPE	MOVE
S_TRANSPRT	TTYPE	PIEC
S_TRANSPRT	TTYPE	TASK
S_TRANSPRT	TTYPE	TRAN

Transaction code
SU53
SE38
SE80
SE91
ST22
AL11
SE10
SE11
SE16
SE37
SM58
SM59

Authorizations Required for RKM Execution

The following authorizations are required for running a reverse-engineering using the **RKM SAP ERP**.

Object	Field Name	Value
S_ADMI_FCD	S_ADMI_FCD	*
S_ADMI_FCD	S_ADMI_FCD	NADM
S_ADMI_FCD	S_ADMI_FCD	PADM
S_CTS_ADMI	CTS_ADMFCT	EPS1
S_CTS_ADMI	CTS_ADMFCT	EPS2
S_CTS_ADMI	CTS_ADMFCT	PROJ
S_C_FUNCT	ACTVT	16
S_C_FUNCT	CFUNCNAME	*
S_C_FUNCT	PROGRAM	*
S_CTS_ADMI	CTS_ADMFCT	TABL
S_TCODE	TCD	SU53
S_TCODE	TCD	SE38
S_TCODE	TCD	SE80
S_DEVELOP	ACTVT	16
S_DATASET	ACTVT	33
S_DATASET	FILENAME	*
S_TCODE	TCD	SE91
S_TCODE	TCD	ST22
S_DATASET	PROGRAM	*
S_RFC	RFC_TYPE	*
S_RFC	RFC_NAME	*
S_RFC	ACTVT	16
S_DEVELOP	ACTVT	3
S_DEVELOP	DEVCLASS	*
S_DEVELOP	OBJNAME	*
S_DEVELOP	OBJTYPE	*
S_DEVELOP	P_GROUP	*
S_DOKU_AUT	DOKU_ACT	RAW_VERS
S_DOKU_AUT	DOKU_DEVCL	TEST
S_DOKU_AUT	DOKU_MODE	MAINTAIN

S_RFC_ADM	ACTVT	3
S_RFC_ADM	ICF_VALUE	*
S_RFC_ADM	RFCDEST	*
S_RFC_ADM	RFCTYPE	*
S_RZL_ADM	ACTVT	3
S_TABU_DIS	ACTVT	3
S_TABU_DIS	DICBERCLS	*
S_TCODE	TCD	AL11
S_TCODE	TCD	SE10
S_TCODE	TCD	SE11
S_TCODE	TCD	SE16
S_TCODE	TCD	SE37
S_TCODE	TCD	SM58
S_TCODE	TCD	SM59
S_TRANSPRT	ACTVT	3
S_TRANSPRT	TTYE	CUST
S_TRANSPRT	TTYE	DTRA
S_TRANSPRT	TTYE	MOVE
S_TRANSPRT	TTYE	PIEC
S_TRANSPRT	TTYE	TASK
S_TRANSPRT	TTYE	TRAN

Authorizations Required for LKM Execution

The following authorizations are required for using the **LKM SAP ERP to Oracle (SQLDR)**. This set of authorization is needed to install the ABAP code generated by the LKM and execute it.

Object	Field Name	Value
S_ADMI_FCD	S_ADMI_FCD	*
S_ADMI_FCD	S_ADMI_FCD	NADM
S_ADMI_FCD	S_ADMI_FCD	PADM
S_CTS_ADMI	CTS_ADMFCT	EPS1
S_CTS_ADMI	CTS_ADMFCT	EPS2

S_CTS_ADMI	CTS_ADMFCT	PROJ
S_C_FUNCT	ACTVT	16
S_C_FUNCT	CFUNCNAME	*
S_C_FUNCT	PROGRAM	*
S_CTS_ADMI	CTS_ADMFCT	TABL
S_TCODE	TCD	SU53
S_TCODE	TCD	SE38
S_TCODE	TCD	SE80
S_DEVELOP	ACTVT	16
S_DATASET	ACTVT	33
S_DEVELOP	ACTVT	7
S_DATASET	FILENAME	*
S_DEVELOP	ACTVT	1
S_TCODE	TCD	SE91
S_TCODE	TCD	ST22
S_DEVELOP	ACTVT	40
S_DEVELOP	ACTVT	41
S_DATASET	ACTVT	34
S_DEVELOP	ACTVT	94
S_DEVELOP	ACTVT	2
S_TRANSPRT	ACTVT	2
S_TRANSPRT	ACTVT	1
S_TRANSPRT	ACTVT	60
S_TABU_DIS	ACTVT	2
S_RZL_ADM	ACTVT	1
S_DATASET	PROGRAM	*
S_RFC	RFC_TYPE	*
S_RFC	RFC_NAME	*
S_RFC	ACTVT	16
S_DEVELOP	ACTVT	3
S_DEVELOP	DEVCLASS	*

S_DEVELOP	OBJNAME	*
S_DEVELOP	OBJTYPE	*
S_DEVELOP	P_GROUP	*
S_DOKU_AUT	DOKU_ACT	RAW_VERS
S_DOKU_AUT	DOKU_DEVCL	TEST
S_DOKU_AUT	DOKU_MODE	MAINTAIN
S_RFC_ADM	ACTVT	3
S_RFC_ADM	ICF_VALUE	*
S_RFC_ADM	RFCDEST	*
S_RFC_ADM	RFCTYPE	*
S_RZL_ADM	ACTVT	3
S_TABU_DIS	ACTVT	3
S_TABU_DIS	DICBERCLS	*
S_TCODE	TCD	AL11
S_TCODE	TCD	SE10
S_TCODE	TCD	SE11
S_TCODE	TCD	SE16
S_TCODE	TCD	SE37
S_TCODE	TCD	SM58
S_TCODE	TCD	SM59
S_TRANSPRT	ACTVT	3
S_TRANSPRT	TTYPE	CUST
S_TRANSPRT	TTYPE	DTRA
S_TRANSPRT	TTYPE	MOVE
S_TRANSPRT	TTYPE	PIEC
S_TRANSPRT	TTYPE	TASK
S_TRANSPRT	TTYPE	TRAN

Authorizations Required for LKM Execution for Production

The following authorizations are required for using the **LKM SAP ERP to Oracle (SQLDR)** in a production environment. In such environment, the ABAP code generated by the LKM is not installed by the LKM but simply executed.

Object	Field Name	Value
S_ADMI_FCD	S_ADMI_FCD	*
S_ADMI_FCD	S_ADMI_FCD	NADM
S_ADMI_FCD	S_ADMI_FCD	PADM
S_CTS_ADMI	CTS_ADMFCT	EPS1
S_CTS_ADMI	CTS_ADMFCT	EPS2
S_CTS_ADMI	CTS_ADMFCT	PROJ
S_C_FUNCT	ACTVT	16
S_C_FUNCT	CFUNCNAME	*
S_C_FUNCT	PROGRAM	*
S_CTS_ADMI	CTS_ADMFCT	TABL
S_TCODE	TCD	SU53
S_TCODE	TCD	SE38
S_TCODE	TCD	SE80
S_DEVELOP	ACTVT	16
S_DATASET	ACTVT	33
S_DATASET	FILENAME	*
S_TCODE	TCD	SE91
S_TCODE	TCD	ST22
S_DATASET	PROGRAM	*
S_RFC	RFC_TYPE	*
S_RFC	RFC_NAME	*
S_RFC	ACTVT	16
S_DEVELOP	ACTVT	3
S_DEVELOP	DEVCLASS	*
S_DEVELOP	OBJNAME	*
S_DEVELOP	OBJTYPE	*
S_DEVELOP	P_GROUP	*
S_DOKU_AUT	DOKU_ACT	RAW_VERS

S_DOKU_AUT	DOKU_DEVCL	TEST
S_DOKU_AUT	DOKU_MODE	MAINTAIN
S_RFC_ADM	ACTVT	3
S_RFC_ADM	ICF_VALUE	*
S_RFC_ADM	RFCDEST	*
S_RFC_ADM	RFCTYPE	*
S_RZL_ADM	ACTVT	3
S_TABU_DIS	ACTVT	3
S_TABU_DIS	DICBERCLS	*
S_TCODE	TCD	AL11
S_TCODE	TCD	SE10
S_TCODE	TCD	SE11
S_TCODE	TCD	SE16
S_TCODE	TCD	SE37
S_TCODE	TCD	SM58
S_TCODE	TCD	SM59
S_TRANSPRT	ACTVT	3
S_TRANSPRT	TTYPE	CUST
S_TRANSPRT	TTYPE	DTRA
S_TRANSPRT	TTYPE	MOVE
S_TRANSPRT	TTYPE	PIEC
S_TRANSPRT	TTYPE	TASK
S_TRANSPRT	TTYPE	TRAN

Appendix B - SAP Stand-Alone Connection Test

In addition to the Connection Testing described above, a test can be performed outside of ODI using a standalone java utility:

To use a standalone java utility to test the connection:

1. Open a command window.
2. Change to `oracledi/drivers` directory.
3. Make sure that `JAVA_HOME` points to a JVM (at least 1.5).

4. Make sure that you have installed SAP Java Connector and that the `sapjco3.jar` and the `sapjco3` library are in the `oracledi/drivers` directory.

5. Launch the utility using the following command:

```
java -cp sapjco3.jar;odi-sap.jar oracle.odi.sap.km.test.JCoTest
```

This should result in an output similar to this:

```
JCO3 Libraray available!!
JCO Version: 3.0.2 (2009-03-29)
Creating connection Pool...
Error: com.sap.conn.jco.JCoException: (102) RFC_ERROR_COMMUNICATION: Connect to SAP gateway failed
Connection parameters: TYPE=A DEST=ODI_SAP_CON_POOL ASHOST=<SAP Application Server> SYSNR=00 PCS=1

LOCATION      CPIC (TCP/IP) on local host with Unicode
ERROR        hostname '<SAP Application Server>' unknown
TIME         Tue Sep 08 16:07:27 2009
RELEASE      711
COMPONENT    NI (network interface)
VERSION      39
RC           -2
MODULE       ninti.c
LINE         895
DETAIL       NiPGetHostByName: '<SAP Application Server>' not found
SYSTEM CALL  getaddrinfo
COUNTER      2
```

6. Use a text editor to open the `ODI_SAP_CON_POOL.jcoDestination` file in the `oracledi/drivers` directory. The file should look like this:

```
#for tests only!
jco.client.lang=EN
jco.destination.peak_limit=10
jco.client.client=800
jco.client.passwd=<SAP Password>
jco.client.user=<SAP User>
jco.client.sysnr=00
jco.destination.pool_capacity=5
jco.client.ashost=<SAP Application Server>
```

7. Enter you SAP connection information, which you have received from your SAP administrator. The file will then look similar to this (use your connection information!):

```
#for tests only !
jco.client.lang=EN
jco.destination.peak_limit=10
jco.client.client=800
jco.client.passwd=ODI123
jco.client.user=ODI
jco.client.sysnr=00
jco.destination.pool_capacity=5
jco.client.ashost=123.123.123.123.
```

8. Launch the utility again using the following command:

```
java -cp sapjco3.jar;odi-sap.jar oracle.odi.sap.km.test.JCoTest
```

This should result in an output similar to this:

```
C:\WINDOWS\system32\cmd.exe
SYSTEM CALL getaddrinfo
COUNTER 2

D:\OraHome_1\oracledi\drivers>java -cp odi-sap.jar;sapjco3.jar oracle.odi.sap.km.test.JCoTest
JCO3 Libraray available!!
JCO Version: 3.0.2 (2009-03-29)
Creating connection Pool...
R3 Name: null
Attributes: DEST: ODI_SAP_CON_POOL
OWN_HOST:
PARTNER_HOST:
SYSTNR: 00
SYSID: NRI
CLIENT: 800
USER:
LANGUAGE: E
ISO_LANGUAGE: EN
OWN_CODEPAGE: 1100
OWN_CHARSET: ISO8859_1
OWN_ENCODING: iso-8859-1
OWN_BYTES_PER_CHAR: 1
PARTNER_CODEPAGE: 1100
PARTNER_CHARSET: ISO8859_1
PARTNER_ENCODING: iso-8859-1
PARTNER_BYTES_PER_CHAR: 1
OWN_REL: 711
PARTNER_REL: 46C
PARTNER_TYPE: 3
KERNEL_REL: 46D
TRACE:
RFC_ROLE: C
OWN_TYPE: E
CPIC_CONUID: 00000000

Successfully Connected to SAP System...
Z_ODI_RFC_GET_TABLES is exist in specified SAP System...
Z_ODI_RFC_READ_TABLE is exist in specified SAP System...
Z_ODI_RFC_TABLE_KEYS is exist in specified SAP System...
Z_ODI_RFC_GET_TABLE_INDEXES is exist in specified SAP System...

D:\OraHome_1\oracledi\drivers>
```

In addition to just testing the SAP connection, the utility will also validate the existence of certain Function Modules required for the RKM. These are installed during first execution of the RKM (UPLOAD_ABAP = Yes).

9. Delete the ODI_SAP_CON_POOL.jcoDestination file after execution, as it contains the SAP login credentials.

Appendix C - Uninstalling ODI SAP Components

In case of upgrading to a newer ODI SAP Connector release or for permanently removing ODI connectivity, please follow the steps described in this section.

Please note that this will delete ALL components including generated extractor programs.

To upgrade to a newer ODI SAP Connector release or to permanently remove the ODI connectivity:

1. Start the SAP GUI.
2. Connect to the SAP systems you want to uninstall.
3. Go to the transaction SE80.
4. Select the package or development class option from the drop down box.
5. Enter ZODI_DEVCLASS in the package or development class field.
6. Right-click the ZODI_DEVCLASS object below the object name tab.
7. Select **Delete** in the drop down menu and delete the development class.

8. If it does not allow deleting the development class individually, delete all the objects one by one and then delete the development class.
9. Go to transaction `SE10`. Select **Modifiable** and **Workbench Requests**. Click **Display**.
10. Select the transport requests which have the ODI Objects and Release them.
11. Now repeat the steps 3 to 7 to delete Development Class.
12. Go to transaction `SE01`. Select **Modifiable** and **Workbench Requests**. Click **Display**. Release the Transport Request which has `ZODI_DEVCLASS`.