

Oracle® Fusion Middleware

Getting Started with SAP ABAP ERP Adapter for Oracle Data Integrator



12c (12.2.1.3.0)

E96501-04

March 2019

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

ORACLE®

Oracle Fusion Middleware Getting Started with SAP ABAP ERP Adapter for Oracle Data Integrator, 12c (12.2.1.3.0)

E96501-04

Copyright © 2010, 2019, Oracle and/or its affiliates. All rights reserved.

Primary Author: Oracle Corporation

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	
	Audience	vi
	Documentation Accessibility	vi
	Related Documents	vi
	Conventions	vii
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
	Requesting ODI SAP Component Installation into the SAP System	2-2
	Requesting FTP Server Access	2-2
	Gathering SAP Connection Information	2-3
	Validating the Shared Folder Setup	2-4
	Validating the FTP Setup	2-4
	Validating SAP Privileges	2-5
	Validating SAP Transport Layer Name	2-6
	Installing Oracle Data Integrator	2-6
	Installing and Configuring the Oracle DB Target	2-7
	Installing and Configuring JCo	2-7
	Configuring Oracle Data Integrator	2-8
	Add the Open Tool	2-8
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	5-7
	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	6-5
	Review the Mapping Execution	6-5
	Review the Resulting Data	6-6
7	Using Advanced Join Handling for Change Detection with SAP CDHDR Table	
	Required Tools and Versions	7-1
	Creating Variables	7-2
	Creating Mapping with Advanced Join Condition	7-4
	Using Filters Condition in Mapping	7-6
	Creating a Package for Extracting Updated Records	7-8
	Understanding Last Extract Timestamp Parameter	7-9
	Running the Package	7-9
	Execute Scenario to Extract all the Updated Records	7-10

Create New Records in SAP table and Execute Package to Extract New Records	7-13
Update Records in SAP and Execute the package to Extract Updated Records	7-17
Delete Records from SAP and Execute Package	7-20

8 Going Further with Oracle Data Integrator

Going further with Oracle Data Integrator	8-1
---	-----

A SAP ABAP ERP Required Privileges

Important points to consider	A-1
Authorizations Required for RKM SAP ERP Upload	A-2
Authorizations Required for RKM SAP ERP Execution	A-3
Authorizations Required for LKM SAP ERP Upload	A-3
Authorizations Required for LKM SAP ERP Execution	A-4

B SAP Stand-Alone Connection Test

SAP Stand-Alone Connection Test	B-1
---------------------------------	-----

C Uninstalling ODI SAP Components

Updating ODI SAP Components	C-1
Uninstalling ODI SAP Components	C-1

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 *Data Integrator Getting Started*.

It is recommended that you also review the SAP ABAP chapter of *Application Adapters Guide for 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 Developer's Guide*
- *Migrating From Oracle Warehouse Builder to Oracle Data Integrator*
- *Oracle Data Integrator Tools 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.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

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

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 (SQLDR)** 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.

2

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

Before you begin, consider the following:

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 *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 or above.

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

Requesting ODI SAP Component Installation into the SAP System

The ODI SAP adapter communicates with the SAP System using a few ODI SAP RFCs. These RFCs are installed by your SAP Basis team using SAP Transport requests. Please contact your SAP administrators for installing the ODI SAP Components and assigning the required SAP user authorizations by following the instructions given in the Installing ODI SAP Components section in *Application Adapters Guide for Oracle Data Integrator*.

Requesting FTP Server Access

This section applies only if you plan to transfer data using FTP. You can skip this section if you use a Shared Directory for the data transfer or if your SAP system is older than the following:

- SAP ECC6 EHP6 or later
- SAP BW/BI 7.4 or later

The following steps must be performed before any subsequent steps in this guide can be performed. These steps are typically performed by your SAP Basis team. Failure to perform these setup instructions will lead to FTP failure when running any ODI SAP extraction jobs.

To request FTP server access:

1. Login to SAP server.
2. Execute transaction SE16.

3. Enter `SAPFTP_SERVERS` in the **Table Name** field and click  icon to create entries.
4. Enter IP address of the FTP server in the **FTP_SERVER_NAME** field.
5. Enter port of the FTP server in the **FTP_Server_Port** field.
6. Click  icon to save.

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.

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 `SAPFTP`.
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 *Data Integrator Getting Started* 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);

CREATE TABLE ODI_SAP_DEMO.MARA_REPLICATED
(
    MANDT VARCHAR2(3 CHAR),
    MATNR VARCHAR2(18 CHAR),
    DELETED VARCHAR2(1 CHAR),
    MEINS VARCHAR2(3 CHAR),
    CDHDR_OBJECTID VARCHAR2(90 CHAR),
    UDATE DATE,
    TCODE VARCHAR2(20 CHAR)
);
```

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.

 **Note:**

- 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

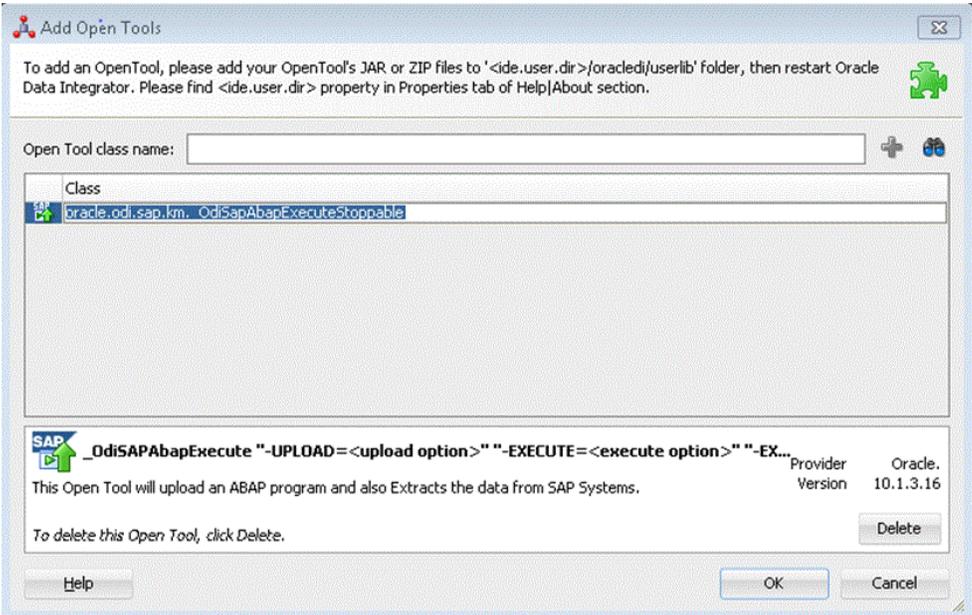
Perform the following steps to configure Oracle Data Integrator:

Add the Open Tool

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

1. Connect to Designer.
2. Select **ODI > 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**.

3

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

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* guide, 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 section of *Connectivity and Knowledge Modules Guide for Oracle Data Integrator Developer's*. Please note 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) section of *Application Adapters Guide for Oracle Data Integrator Guide*.

- **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 (SQLDR)/ 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.

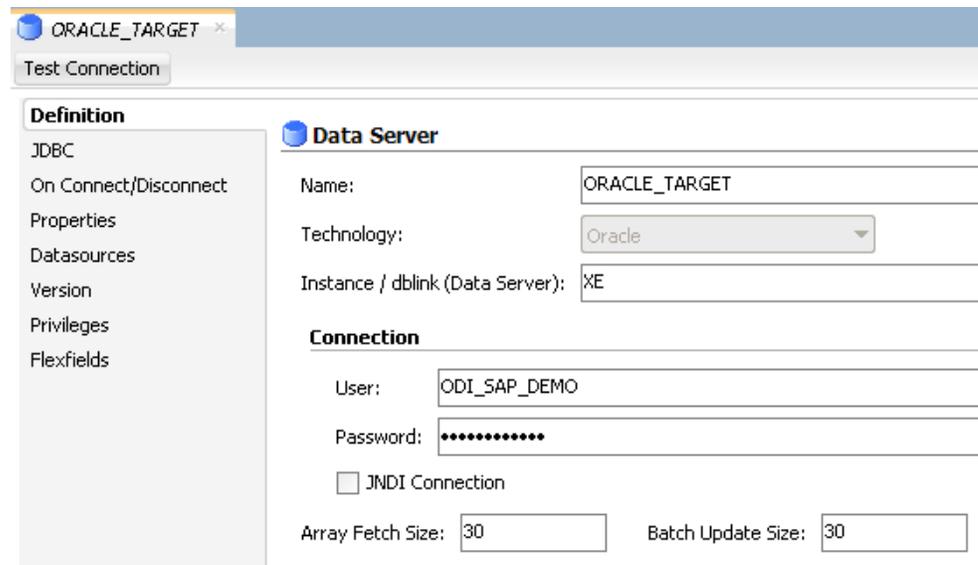
Refer to *File Transfer Considerations* section of *Application Adapters Guide for Oracle Data Integrator Guide*.

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.

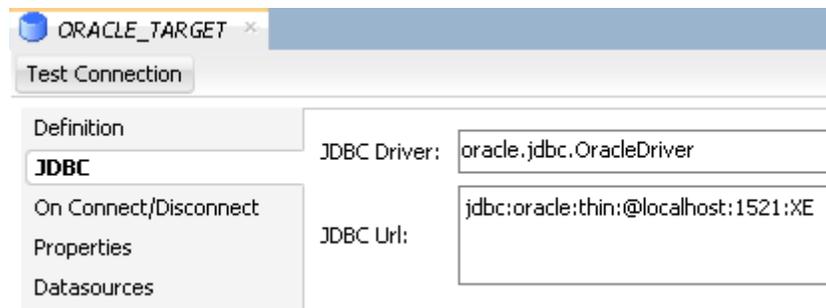
Configuring the Target Data Server

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

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 [Gathering SAP Connection Information](#) 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, EE8IS08859P2 for Croatian Data. For UNICODE systems, use UTF8.

Note:

This FlexField is only used with "LKM SAP BW to Oracle (SQLDR)". For use with "LKM SAP BW 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](#) 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.

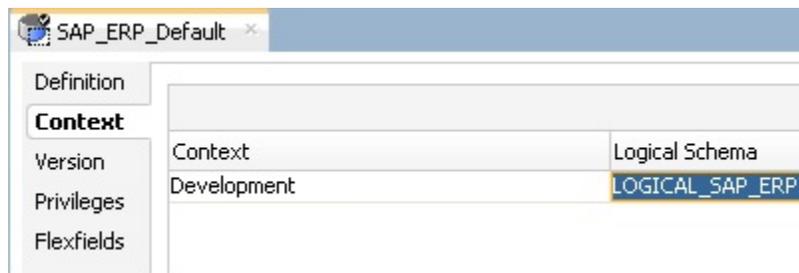
5. From the **File** menu, click **Save**.

 **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.

5

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

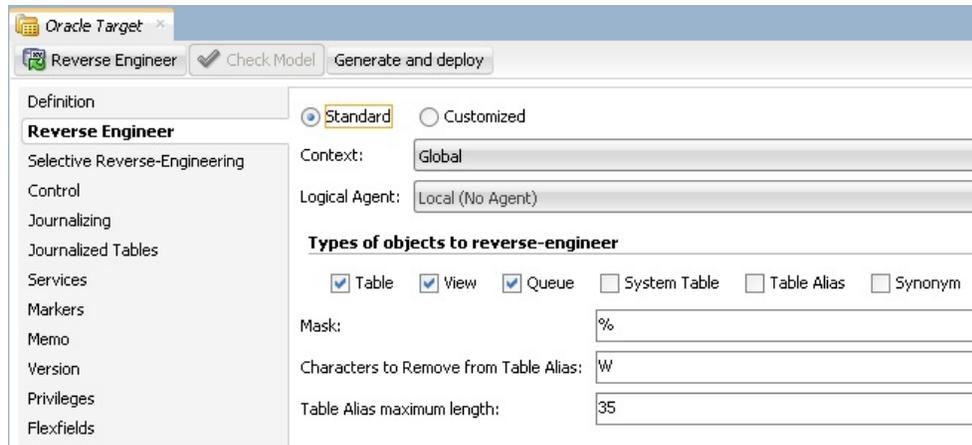
To reverse-engineer the Oracle data model:

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:

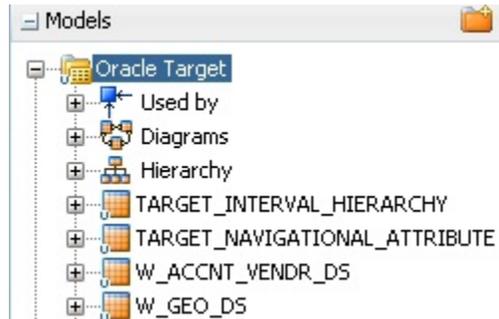
The screenshot shows the 'Oracle Target' dialog box in the 'Definition' tab. The 'Reverse Engineer' button is active. The 'Model' section contains the following fields:

Name:	Oracle Target
Code:	ORACLE_TARGET
Technology:	Oracle
Logical Schema:	ODI_SAP_DEMO
Action Group:	<<Generic Action>>
Default Folder:	
<input type="checkbox"/> Display the Metadata changes in the Model tree	
Description:	

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.



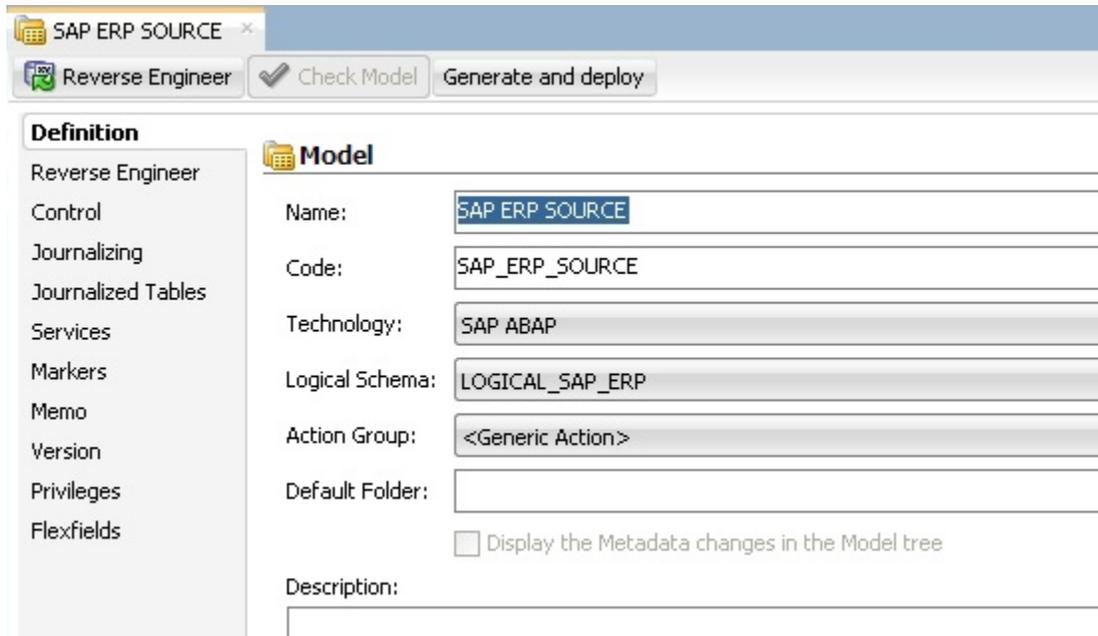
Reverse-Engineering the SAP Source Databases

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

This section also includes the following topics:

Creating the Data Model

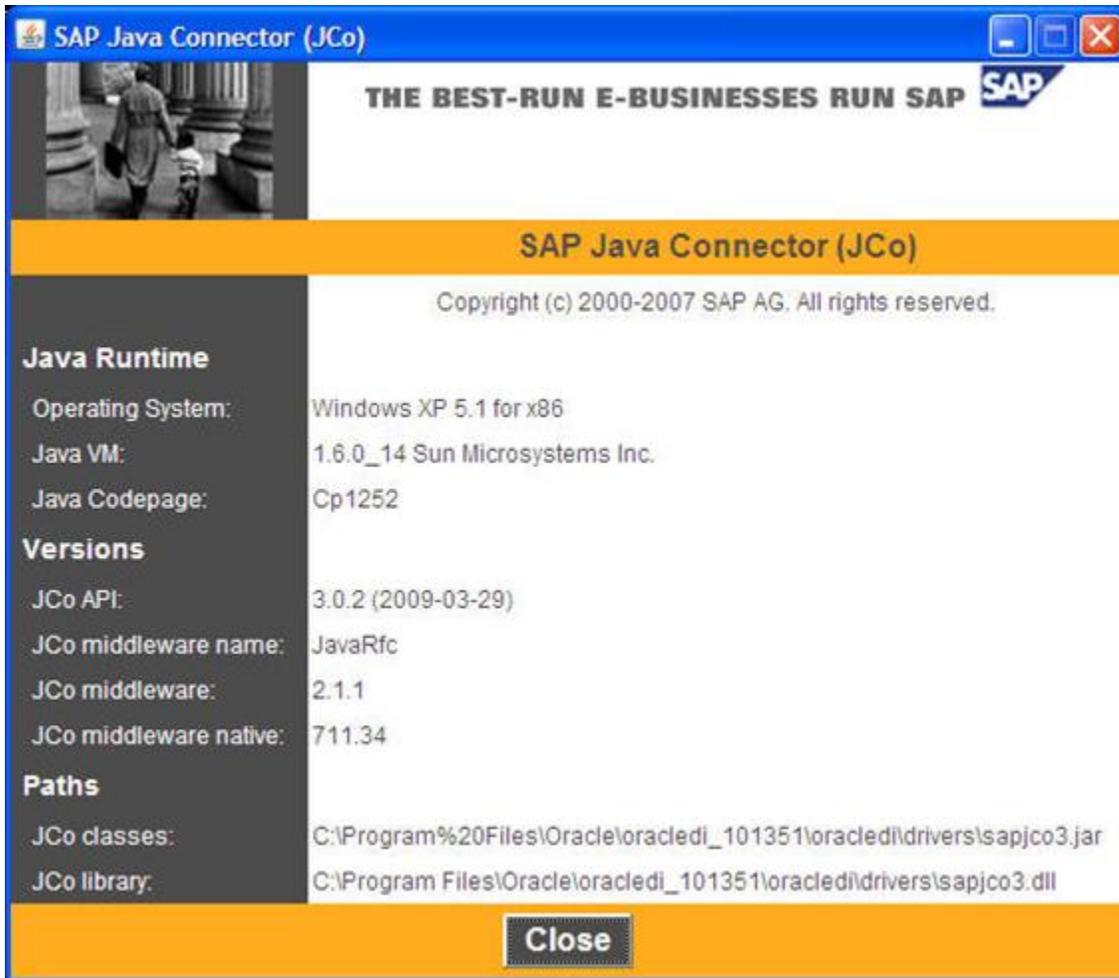
1. In the Models tree view, click **New Model**.
2. In the Definition tab, enter the model parameters as shown below. You must select the SAP ABAP technology and the Logical Schema previously created (in our example it is the LOGICAL_SAP_ERP Logical Schema).



3. In the Reverse Engineer tab:
4.
 - 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](#) 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 [SAP Stand-Alone Connection Test](#).

Starting the Reverse-Engineering Process

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

1. Import TR.

 **Note:**

See also, *Installing SAP Transport Request (TR) in the Application Adapters Guide for Oracle Data Integrator*.

2. In the Models tree view, open the SAP ERP Source Model.
3. 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 `true`.
 - e. Set `UPLOAD_ABAP_CODE` and `UPLOAD_ABAP_BASE` to `False`.

 **Caution:**

Make sure that you provide the correct value for the `SAP_TRANSPORT_LAYER_NAME` option.

- f. Set other parameters according to your SAP configuration. See the *Application Adapters Guide for Oracle Data Integrator* for more information on the RKM options.
4. From the **File** menu, click **Save** to save your changes.
 5. Click **Reverse Engineer** to start the reverse-engineering process.
 6. Click **OK**.
The Sessions Started Dialog is displayed.
 7. Click **OK**.

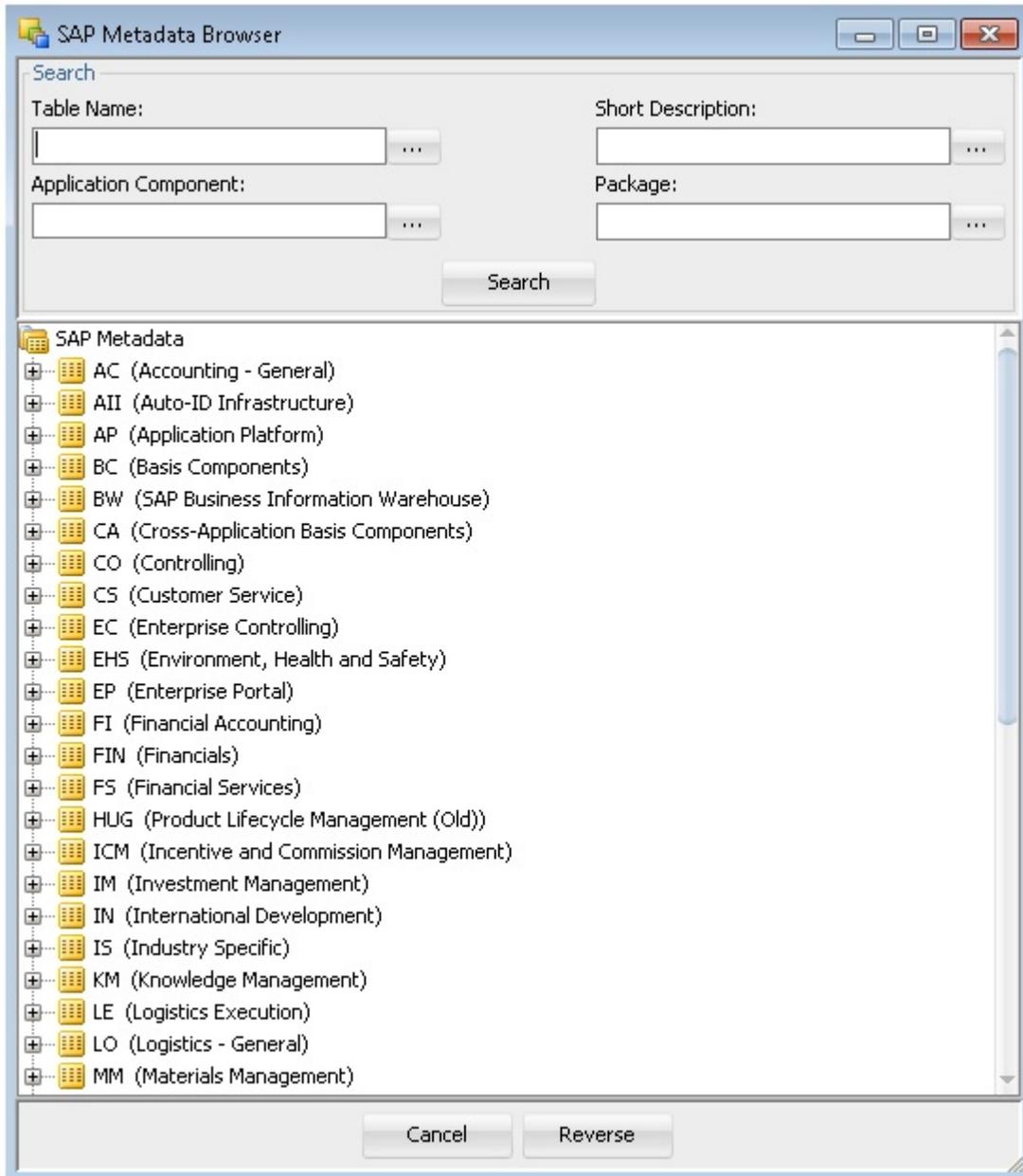
Using the SAP Metadata Browser

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.

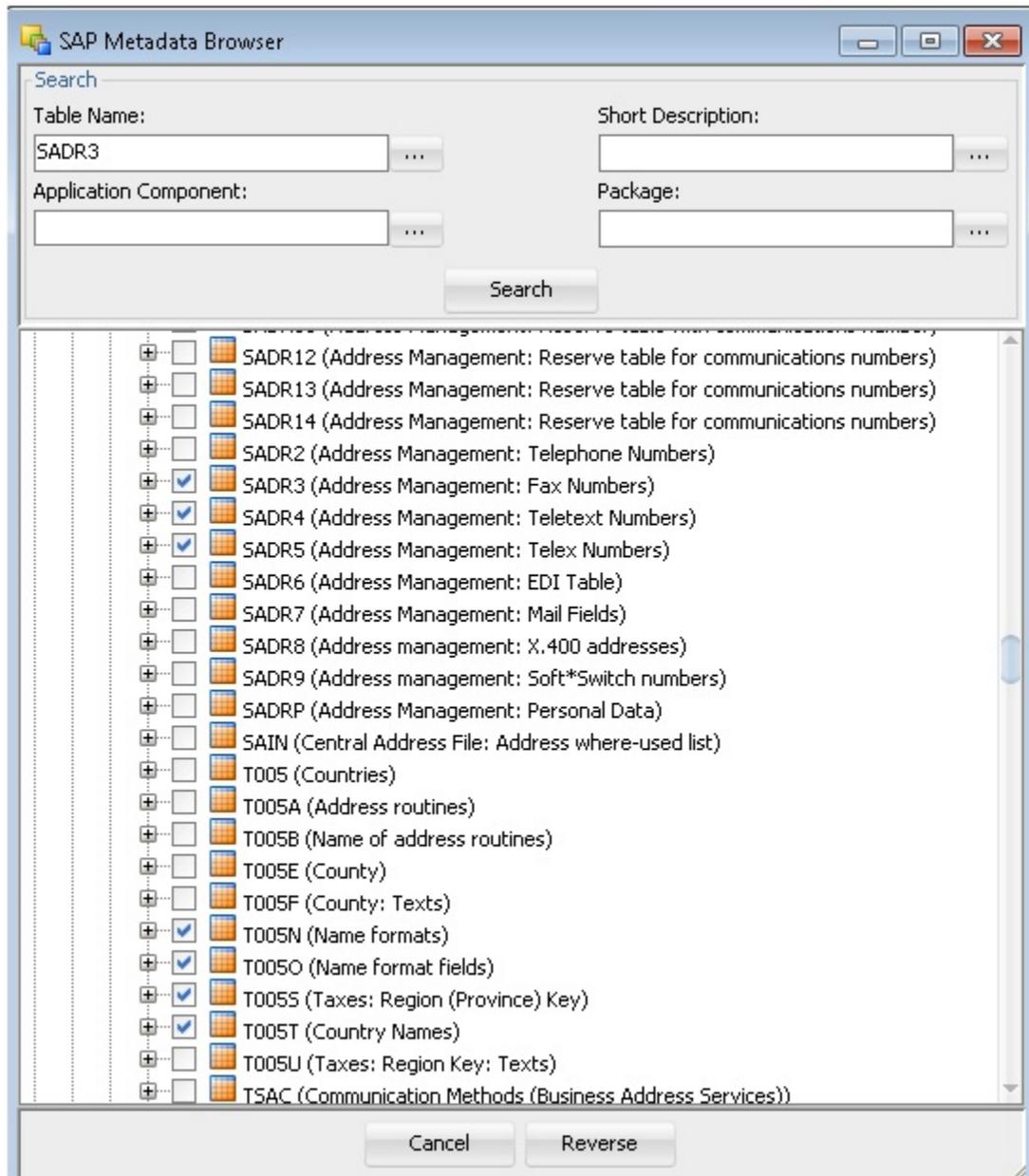
Refer to Log Files in *Application Adapters Guide for Oracle Data Integrator* for more details.



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

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:

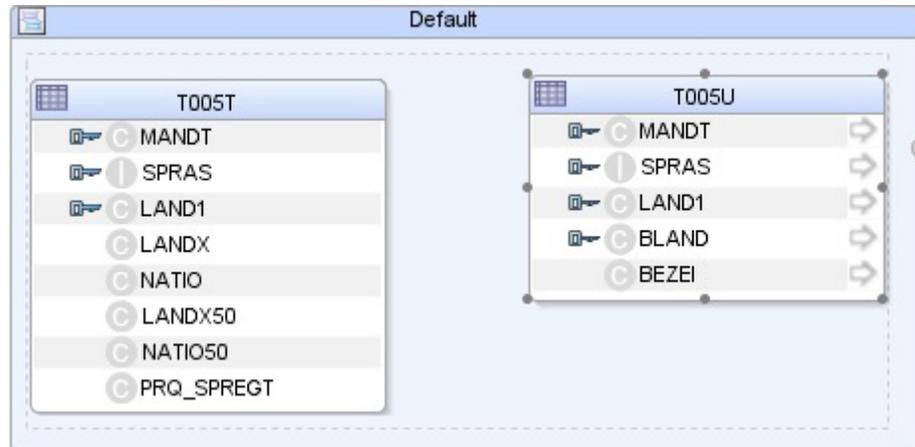
Create the Mapping

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:

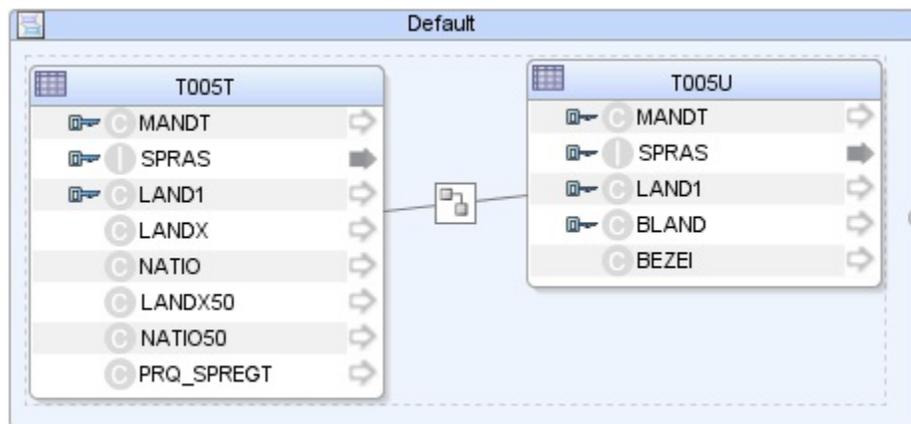
- In the Models tree view, select the WS_GEO_DS datastore from the Oracle Target Model.
- Drag this datastore into the Target Datastore area of the Logical diagram. (Right area of the Logical diagram)
- In the Models tree view, select the T005T Country Names datastore from the SAP ERP Source model.
- 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:



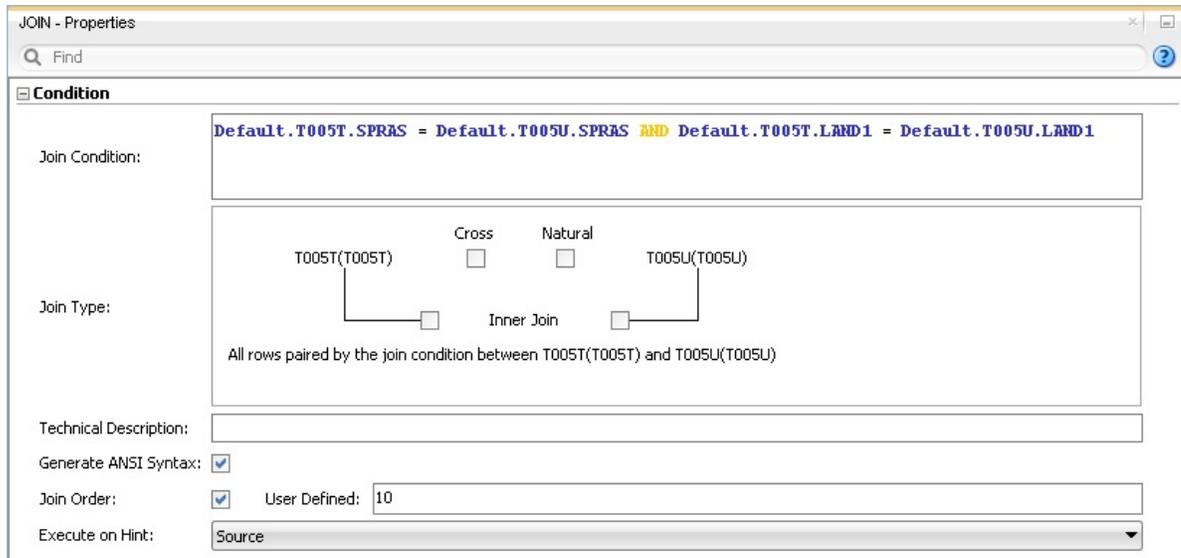
Define Joins between Sources

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:

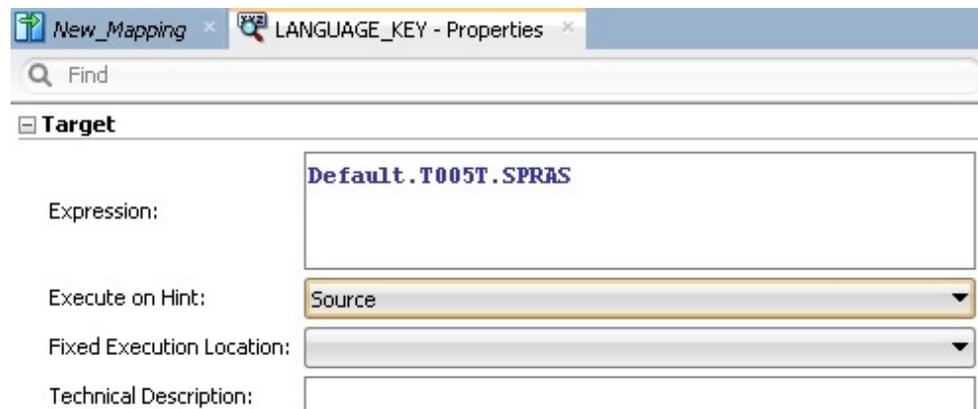


5. Make sure that table number of T005U is smaller than that of T005T. In *1:n relationships* the parent table has to be the first in a join.

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:

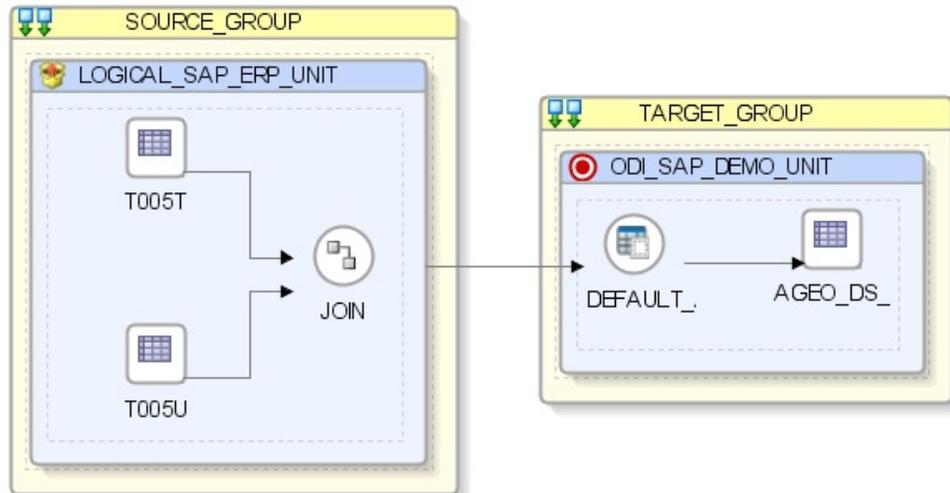


4. Repeat this operation to perform the following simple mappings:
 - W_GEO_DS.COUNTRY = T005T.LAND1
 - W_GEO_DS.COUNTY = T005T.LANDX
 - W_GEO_DS.STATE_PROV = T005U.BLAND
 - W_GEO_DS.NATIONALITY = T005T.NATIO

Define the Mapping Flow

To define the mapping flow:

1. Select the Physical diagram. The Physical 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 (SQLDR) 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

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.

7

Using Advanced Join Handling for Change Detection with SAP CDHDR Table

The SAP system table `CDHDR` holds information about data changes in SAP business tables like `MARA`. Among other data this includes the (concatenated) business key (field `CDHDR.OBJECTID`), change type and change timestamp (field `CDHDR.UPDATE` and `UTIME`).

In this example the business key is the material number (`MARA.MATNR`). Starting with v42 the ODI SAP adapter for SAP ERP/ECC supports substring operations in join conditions. This chapter gives an example of how to use a complex join expression between the SAP tables `MARA` and `CDHDR` to extract only new/modified/deleted `MARA` records.

In case of `MARA` you can identify relevant changes with the help of `CDHDR` by applying these filter/join conditions:

- record type filter on `CDHDR`: `CDHDR.OBJECTCLASS = 'MATERIAL'`
- time filter on `CDHDR`: the timestamp (`CDHDR.UPDATE`, `CDHDR.UTIME`) is greater than the last extraction timestamp and smaller than the current timestamp
- join condition: the first 18 characters of `CDHDR.OBJECTID` match `MARA.MATNR`

Note:

For other object classes and their object id composition/ business key, contact your SAP team.

The following sections cover all the steps needed for building an ODI scenario for delta extraction: on every execution the package extracts any `MARA` changes since last run and applies the changes to the Oracle target table.

- [Setting Up Required Tools and Versions](#)
- [Creating Variables](#)
- [Creating Mapping with Advanced Join Condition](#)
- [Using Filters Condition in Mapping](#)
- [Creating a Package for Extracting Updated Records](#)
- [Understanding Last Extract Timestamp Parameter](#)
- [Running the Package](#)

Required Tools and Versions

This section lists the required tools and their versions.

1. **ODI Version** — Oracle Data Integrator Version 12.2.1

2. **Open Tool Version** — Oracle SAP Connector Version 10.1.3.23
3. **JCo version**
 - a. JCo API — 3.0.17 (2017 -07-17)
 - b. JCo Middleware — JavaRfc 2.2.20
 - c. JCo Library — 721.913
 - d. Path to JCo Archive — c:\Oracle\Middleware\Oracle_Home1\odi\sd\lib\sapjco3.jar
 - e. Path to JCo Library — c:\Oracle\Middleware\Oracle_Home1\odi\sd\lib\sapjco3.dll
4. **SAP KMs Build Number** - 42
5. **IKM** - IKM Oracle Incremental Update

Creating Variables

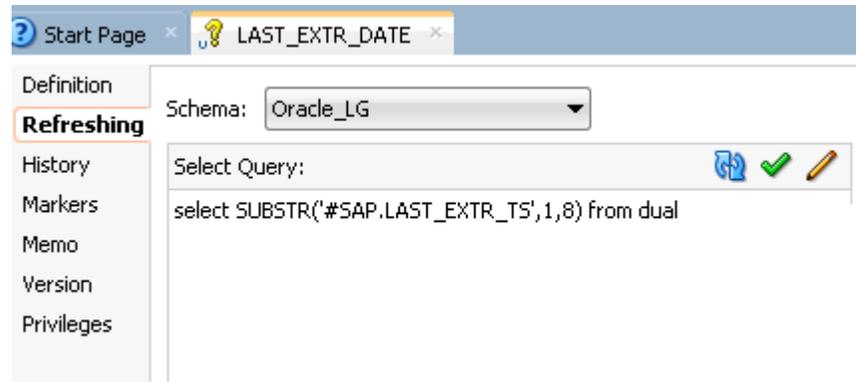
This section describes all the ODI variables used in the below example. Through these variables you can save and get the value of last extract timestamp. Please create all the variables as specified below:

1. **LAST_EXTR_TS** — This variable contains the value for Last Extract Timestamp. Its default value is 00000000:000000 which is required for first attempt and it keeps the latest value in history as shown below:

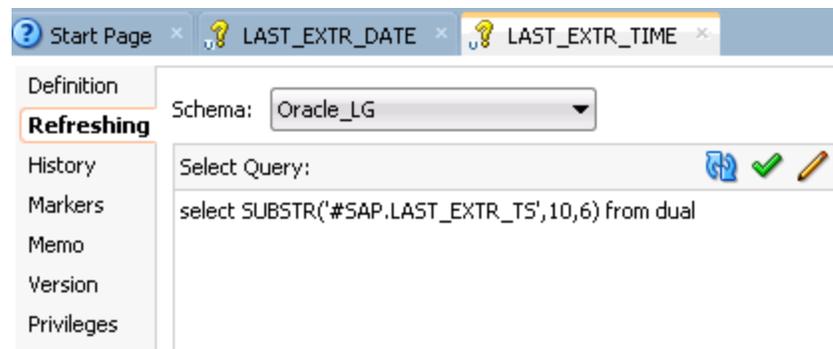
The screenshot shows the 'Definition' tab of the 'Variable [Project: SAP]' window. The variable name is 'LAST_EXTR_TS', its datatype is 'Alphanumeric', and it is set to 'Keep History' with the option 'Latest Value' selected. The 'Secure Value' checkbox is unchecked, and the 'Default Value' is '00000000:000000'. The 'Description' field is empty.

Property	Value
Name:	LAST_EXTR_TS
Datatype:	Alphanumeric
Keep History:	Latest Value
Secure Value:	<input type="checkbox"/>
Default Value:	00000000:000000
Description:	

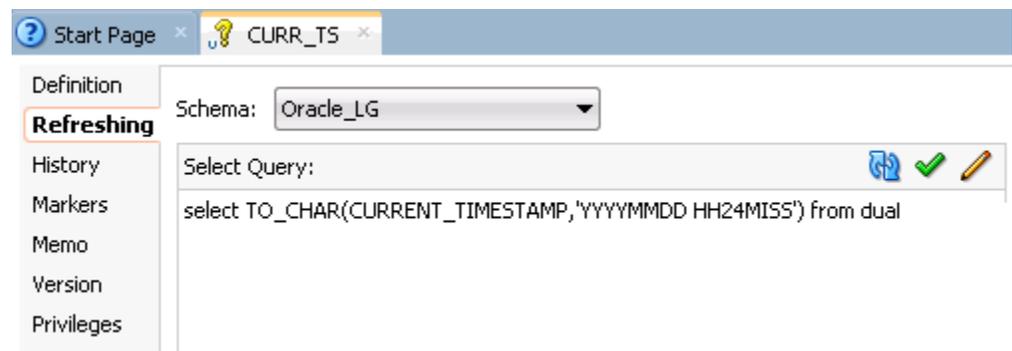
2. **LAST_EXTR_DATE** — This variable contains Last Extract Date which takes its value from the Last Extracted Timestamp parameter. The refresh expression shown below is used to update the variable:



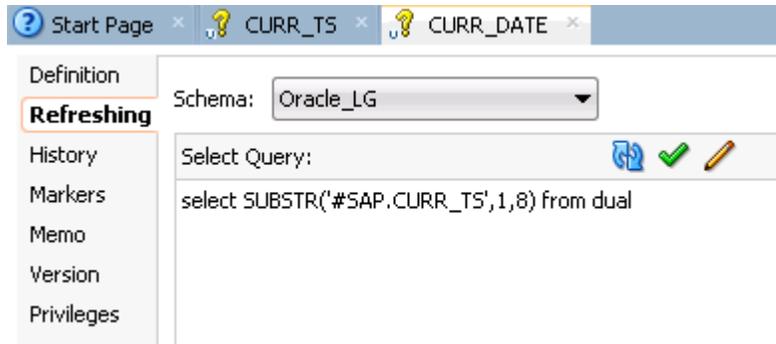
3. **LAST_EXTR_TIME** — This variable contains Last Extract Time which takes its value from Last Extracted Timestamp parameter.



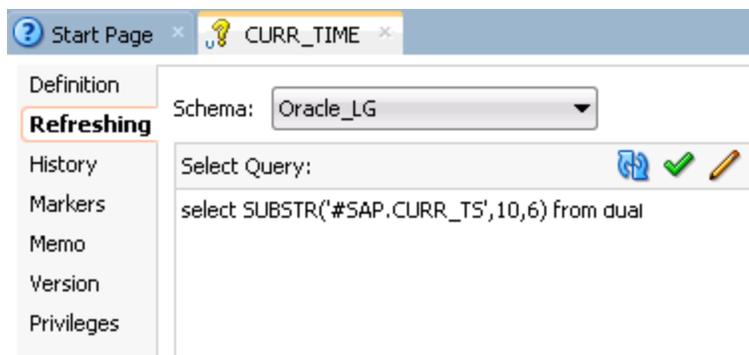
4. **CURR_TS**— This variable contains the Current Timestamp. The refresh expression shown below is used to update the variable:



5. **CURR_DATE**— This variable contains Current Date which takes its value from `CURR_TS` parameter. The refresh expression shown in the below image is used to update the variable:



- CURR_TIME** — This variable contains Current Time which takes its value from CURR_TS parameter. The refresh expression shown below is used to update the variable:



Creating Mapping with Advanced Join Condition

Create a simple mapping where you can join two tables as CDHDR and MARA based on below condition and use two filters to extract the data from SAP and store in target oracle table named MARA_REPLICATED.

- Join Condition**

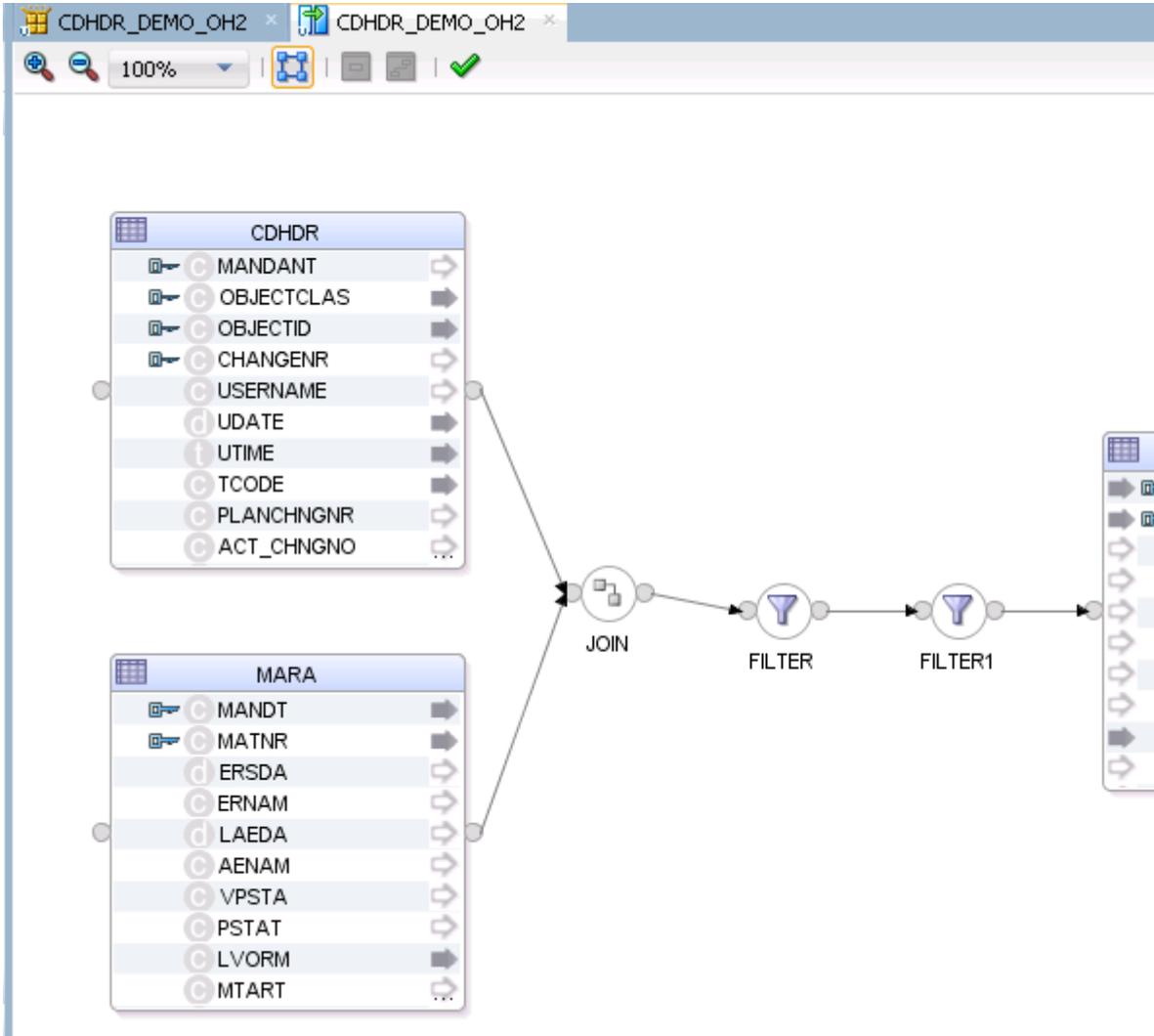
Join OBJECTID of CDHDR from index zero to 18 with MATNR of MARA.

```
CDHDR.OBJECTID+0(18) = MARA.MATNR
```

- Mapped fields from Source to Target**

In this mapping, you can map (by drag and drop) the following fields from source (MARA and CDHDR) to target (MARA_REPLICATED).

```
MARA.MANDT -> MARA_REPLICATED.MANDT
MARA.MATNR -> MARA_REPLICATED.MATNR
MARA.LVORM -> MARA_REPLICATED.DELETED
MARA.MEINS -> MARA_REPLICATED.MEINS
CDHDR.OBJECTID -> MARA_REPLICATED.CDHDR_OBJECTID
CDHDR.UDATE -> MARA_REPLICATED.UDATE
CDHDR.TCODE -> MARA_REPLICATED.TCODE
```



Start Page | JOIN - Properties | CDHDR_DEMO | LAST_EXTR_TS | Session CDHDR_DEMO

Find

Condition

Join Condition: `CDHDR.OBJECTID+0(18) = MARA.MATNR`

Join Type:

Cross Natural
 Left Outer Join

All rows of CDHDR(CDHDR) including the rows unpaired with MARA(MARA) rows

Technical Description: `[CDHDR Change document header (CDHDR)].[Object value]+0(18) = [MARA General Mater`

Using Filters Condition in Mapping

You can use filter conditions to select the related data change log entries. Listed below are two filters — OBJECTCLAS of CDHDR and the next filter is using few variables related to date and time as mentioned above for extracting updated records.

- **Filter 1** contains below condition:

```
CDHDR.OBJECTCLAS = 'MATERIAL'
```

The screenshot shows the 'FILTER - Properties' dialog box in SAP. It is divided into three main sections: Condition, General, and Connector Points.

- Condition:**
 - Filter Condition: `CDHDR.OBJECTCLAS = 'MATERIAL'`
 - Technical Description: `[CDHDR Change document header (CDHDR)].[Object class] = 'MATERIAL'`
 - Execute on Hint: `No Hint`
- General:**
 - Name: `FILTER`
 - Description: (empty)
 - Active:
 - Component Type: `FILTER`
- Connector Points:**
 - Input Connector Points:

Name	Description	Bound Object
INPUT1		

- **Filter 2** contains below condition:

Based on this condition you can retrieve only updated records which is not extracted yet.

```
(CDHDR.UDATE > #SAP.LAST_EXTR_DATE OR (CDHDR.UDATE = #SAP.LAST_EXTR_DATE AND CDHDR.UTIME > #SAP.LAST_EXTR_TIME)) AND (CDHDR.UDATE < #SAP.CURR_DATE OR (CDHDR.UDATE = #SAP.CURR_DATE AND CDHDR.UTIME <= #SAP.CURR_TIME))
```

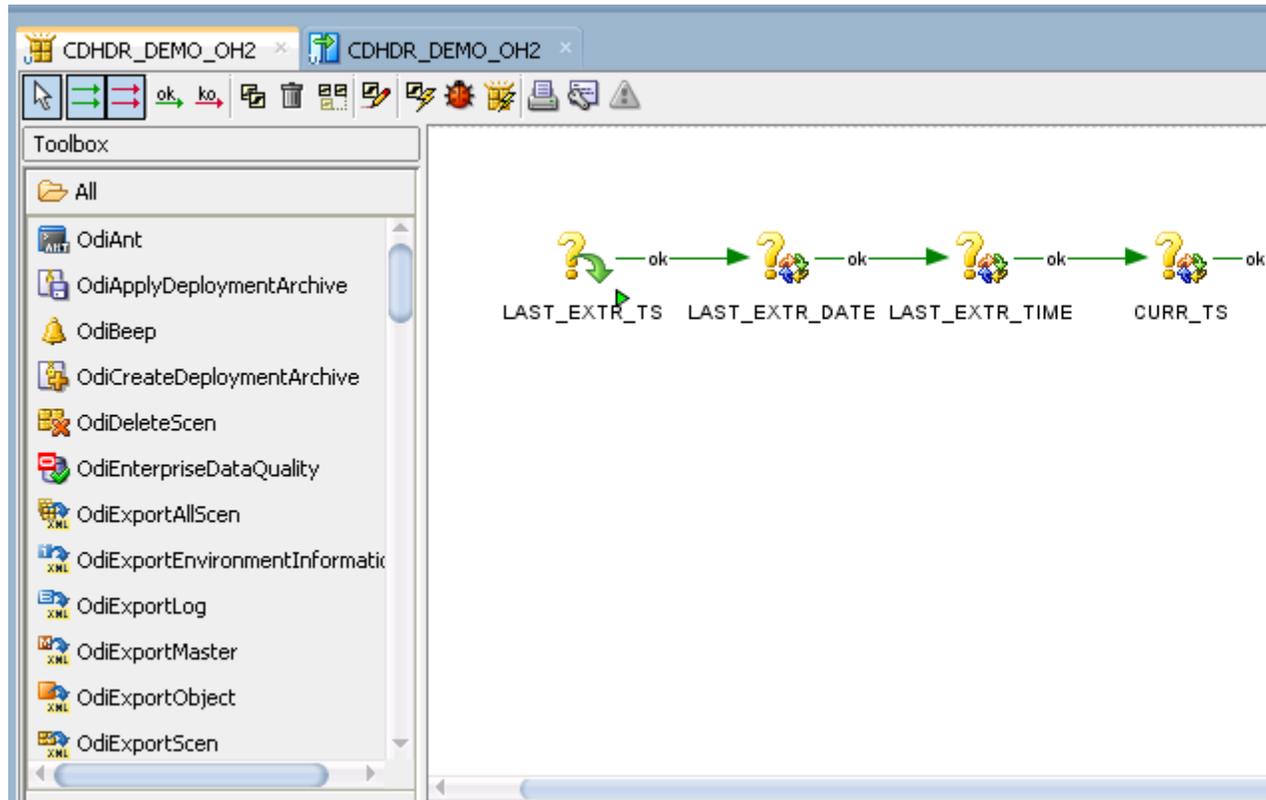
The screenshot shows the Oracle BI Properties window for a filter named FILTER1. The window is divided into several sections:

- Condition:**
 - Filter Condition: `{CDHDR.UDATE > #SAP.LAST_EXTR_DATE OR (CDHDR.UDATE = #SAP.LAST_EXTR_DATE AND CDHDR.UDATE < #SAP.CURR_DATE OR (CDHDR.UDATE = #SAP.CURR_DATE AND CDHDR.UDATE < #SAP.LAST_EXTR_DATE)}`
 - Technical Description: `ader (CDHDR)),[Creation date of the change document] = [SAP.CURR_DATE] AND [CDHDR.UDATE] < [SAP.LAST_EXTR_DATE]`
 - Execute on Hint: No Hint
- General:**
 - Name: FILTER1
 - Description:
 - Active:
 - Component Type: FILTER
- Connector Points:**
 - Input Connector Points:

Name	Description	Bound Object
INPUT1		
 - Output Connector Points:

Creating a Package for Extracting Updated Records

Create a package by following the instructions given below which contains few Date and Time range ODI variables and also create a mapping as described above:



There are eight steps in creating the package (termed as Steps Hierarchy) as shown below:

Blueprint	#	Steps Hierarchy	Step Type
Privileges		CDHDR_DEMO	
	0	LAST_EXTR_TS	Declare Variable
	1	LAST_EXTR_DATE	Refresh Variable
	2	LAST_EXTR_TIME	Refresh Variable
	3	CURR_TS	Refresh Variable
	4	CURR_DATE	Refresh Variable
	5	CURR_TIME	Refresh Variable
	6	CDHDR_MARA_JOIN_2	Mapping
	7	LAST_EXTR_TS	Set Variable

Set Variable: Set the value for LAST_EXTR_TS variable as shown below:

Properties

General | Advanced | Memo | Version | Privileges

Step name
LAST_EXTR_TS

Linked object
LAST_EXTR_TS

Assign Increment

Value
#SAP.CURR_TS

Understanding Last Extract Timestamp Parameter

Before extracting the records, you have to check Last Extract Timestamp parameter so that you can know the exact time when the records were extracted. Default value of Last Extract Timestamp is 00000000:000000 and after first execution it picks its value from History.

Start Page | CDHDR_DEMO | LAST_EXTR_TS | Session CDHDR_DEMO (830)

Definition

Variable [Project: SAP]

Refreshing

History

Markers

Memo

Version

Privileges

Name: LAST_EXTR_TS

Datatype: Alphanumeric

Keep History: Latest Value

Secure Value:

Default Value: 00000000:000000

Description:

Running the Package

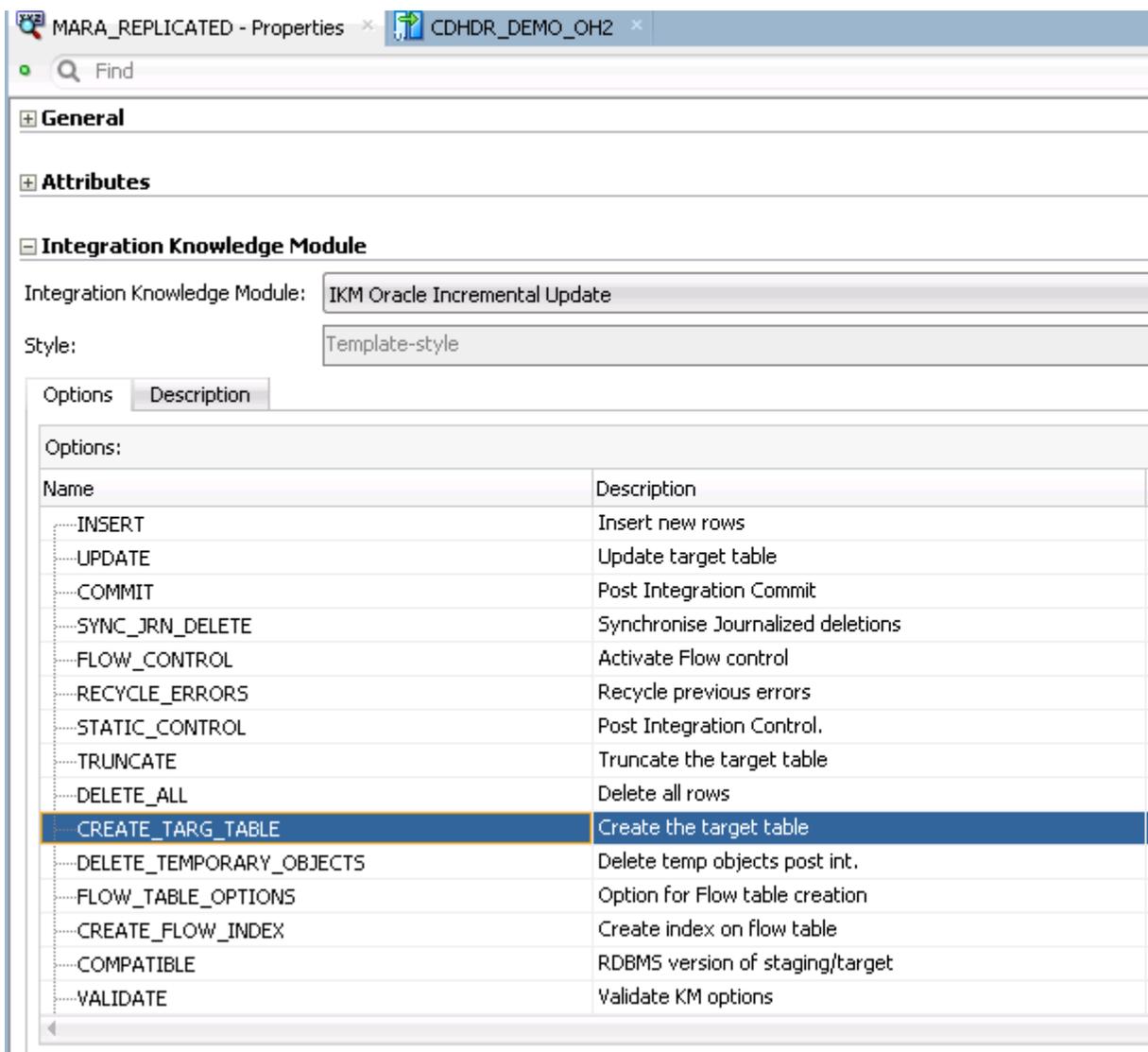
By using Join function with CDHDR table there are various execution processes to extract record from SAP and store in Oracle database. Here we are using V42 SAP KMs with Incremental IKM.

- [Execute Scenario to Extract all the Updated Records](#)
- [Create New Records in SAP table and Execute Package to Extract New Records](#)
- [Update Records in SAP and Execute the package to Extract Updated Records](#)
- [Delete Records from SAP and Execute Package](#)

Execute Scenario to Extract all the Updated Records

In the IKM, set the `Create_Target_Table` option to `True`, as you are retrieving records for the first time, as given below:

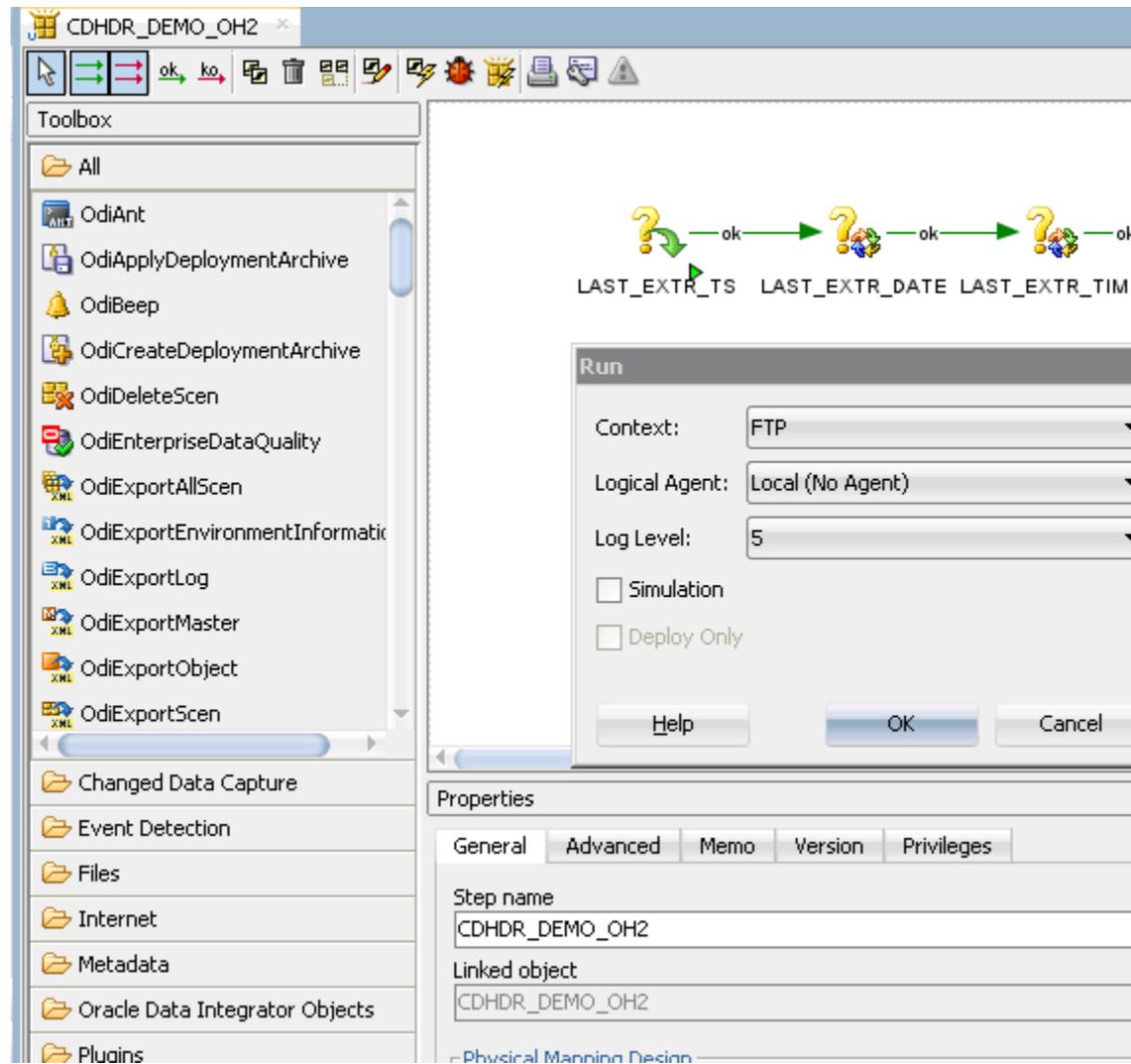
- **Modify the Mapping flow**



The screenshot shows the 'MARA_REPLICATED - Properties' window with the 'CDHDR_DEMO_OH2' scenario selected. The 'Integration Knowledge Module' is set to 'IKM Oracle Incremental Update' and the 'Style' is 'Template-style'. The 'Options' tab is active, displaying a table of options:

Name	Description
INSERT	Insert new rows
UPDATE	Update target table
COMMIT	Post Integration Commit
SYNC_JRN_DELETE	Synchronise Journalized deletions
FLOW_CONTROL	Activate Flow control
RECYCLE_ERRORS	Recycle previous errors
STATIC_CONTROL	Post Integration Control.
TRUNCATE	Truncate the target table
DELETE_ALL	Delete all rows
CREATE_TARG_TABLE	Create the target table
DELETE_TEMPORARY_OBJECTS	Delete temp objects post int.
FLOW_TABLE_OPTIONS	Option for Flow table creation
CREATE_FLOW_INDEX	Create index on flow table
COMPATIBLE	RDBMS version of staging/target
VALIDATE	Validate KM options

- **Execute Package** — Now execute the package to extract all the records which are updated until specific date. When you are executing the package for the first time, it picks its default value from Last Extract Timestamp parameter.



- **Check Extracted Records** — Upon successful execution of the Package, related records are inserted to the Oracle Target Database.

The screenshot displays the Oracle SQL Developer interface. On the left, the 'Session List' pane shows a list of sessions under 'All Executions'. Session 873 is highlighted. The right pane shows the 'Definition' for the selected session, 'Session CDHDR_DEMO_OH2 (873)'. The 'Record Statistics' section is circled in green, showing the following values:

No. of Inserts:	554	No. of Updates:	0
No. of Errors:	0	No. of Rows:	1,662

- **Extracted Records** — Find the extracted records in target database as show below:

```

2
3
4 SELECT MANDT, MATNR, UDATE, TCODE, DELETED, MEINS FROM MARA_REPLICAT
5

```

Data Grid

Messages | Data Grid | Trace | DBMS Output (disabled) | Query Viewer | Explain Plan

MANDT	MATNR	UDATE	TCODE	DELETED	MEINS
810	00000000000000000038	11/19/2018	MM01	X	G
810	00000000000000000037	11/19/2018	MM01	X	%
810	00000000000000000039	11/19/2018	MM01		CCM
810	00000000000000000036	11/5/2018	MM01	X	KAR
810	00000000000000000035	11/5/2018	MM01	X	CCM
810	00000000000000000031	10/29/2018	MM01		%
810	00000000000000000034	10/29/2018	MM01		G
810	00000000000000000033	10/29/2018	MM01		HL
810	00000000000000000032	10/29/2018	MM01		GAU
810	00000000000000000023	7/30/2018	MM01		ST
810	00000000000000000022	7/26/2018	MM01		ST
810	00000000000000000021	7/18/2018	MM01		ST

4: 22 | 720 msec | Row 1 of 500 fetched so far (more rows exist) | DEV42REPO@10.30.32.71:1521/PDE

APS | NUM | INS

Create New Records in SAP table and Execute Package to Extract New Records

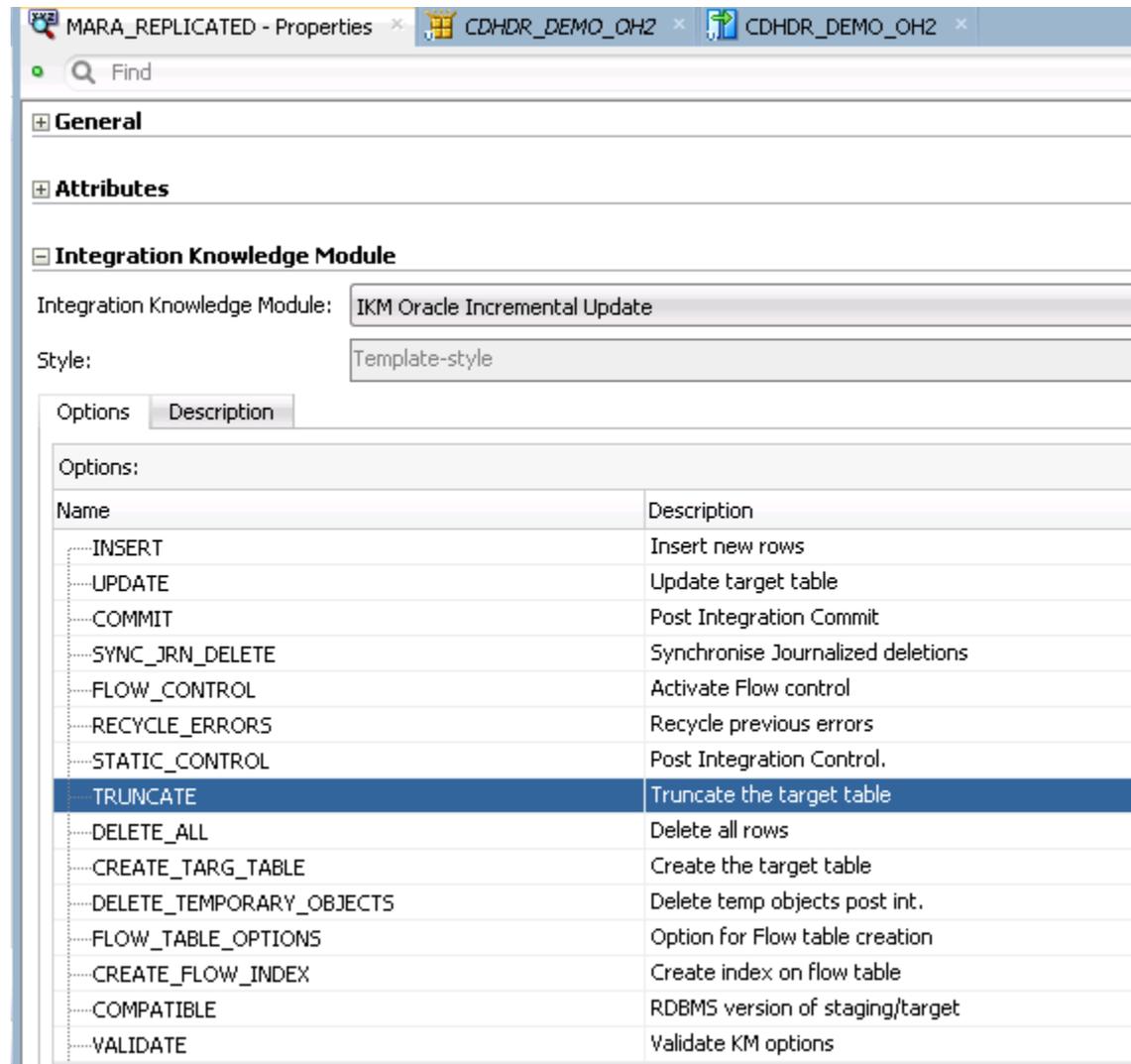
To create new records in SAP table:

1. Create a new material as shown below:

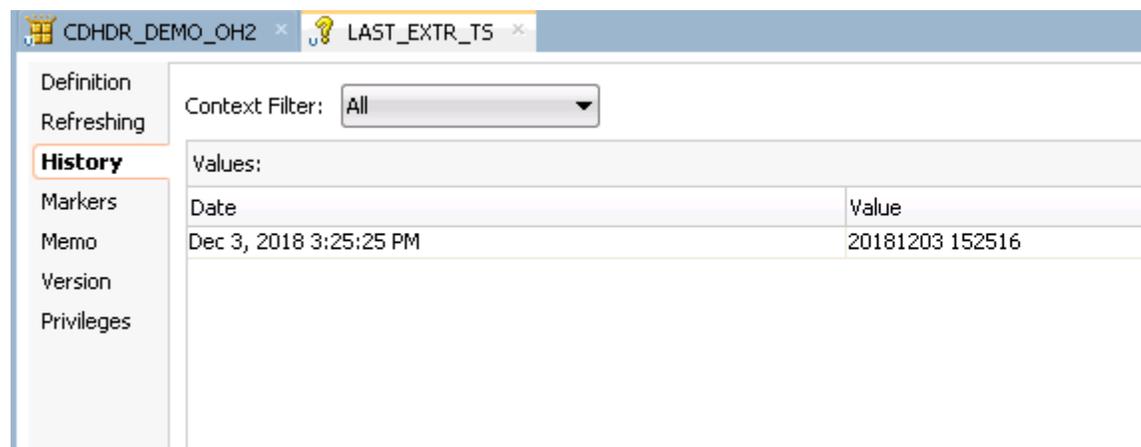
Select View(s)	Org. Levels	Data
Material	<input type="text"/>	<input type="button" value="🔍"/>
Industry sector	Mechanical enginee... ▾	
Material Type	Finished product ▾	
Change Number	<input type="text"/>	
Copy from...		
Material	00000000000000000038	

✔ Material 00000000000000000041 created

2. Modify the Mapping Flow



3. Check the value for last extracted Timestamp in the History tag of LAST_EXTR_TS parameter.



4. Execute the Package - After creating a new material in the SAP table execute the package again. When the Package gets executed successfully, one new record is inserted in the Oracle Target Database.

The screenshot shows the Oracle APEX interface for defining a session. The 'Session' definition is shown with the following details:

- Instance ID: 894
- Session Name: CDHDR_DEMO_OH2
- Scenario Name: (empty)
- Scenario Version: (empty)
- Start: Dec 3, 2018 4:28:03 PM
- Duration (seconds): 9
- Restart Type: no-restart
- URL: (empty)

The 'Concurrent Execution Controller' section is expanded, showing:

- Limit Concurrent Executions
- Violation Behavior: Raise Execution Error
- Wait Polling Interval: (empty)

The 'Record Statistics' section is expanded and circled in green, showing the following statistics:

No. of Inserts:	1	No. of Updates:	0	No. of Deletes:	0
No. of Errors:	0	No. of Rows:	3		

The 'Error Message' field is empty.

5. Extracted Records- The top most records in the below table is the new record which is inserted post package execution.

MANDT	MATNR	UDATE	TCODE	DELETED	MEINS
810	000000000000000041	12/3/2018	MM01		G
810	000000000000000039	11/19/2018	MM01		CCM
810	000000000000000038	11/19/2018	MM01	X	G
810	000000000000000037	11/19/2018	MM01	X	%
810	000000000000000036	11/5/2018	MM01	X	KAR
810	000000000000000035	11/5/2018	MM01	X	CCM
810	000000000000000031	10/29/2018	MM01		%
810	000000000000000034	10/29/2018	MM01		G
810	000000000000000033	10/29/2018	MM01		HL
810	000000000000000032	10/29/2018	MM01		GAU
810	000000000000000023	7/30/2018	MM01		ST
810	000000000000000022	7/26/2018	MM01		ST

Update Records in SAP and Execute the package to Extract Updated Records

From the above image, the value of column MEINS for material 33 and 41 is as HL and G respectively. Now you can update the MEINS column value for these records as shown below:

1. Update the value for these two records in SAP for material number 41 and 33 as shown below:

Figure 7-1 For Material 33

Change Material (Initial Screen)

Select View(s) Org. Levels Data

Material

Change Number

✔ Material 00000000000000000033 changed

Figure 7-2 For Material 41

Change Material (Initial Screen)

Select View(s) Org. Levels Data

Material

Change Number

✔ Material 00000000000000000041 changed

2. Check the last extracted Timestamp from the History tag of the parameter `LAST_EXTR_TS`.

CDHDR_DEMO_OH2 LAST_EXTR_TS

Definition Context Filter: All

Refreshing

History Values:

Date	Value	Context
Dec 3, 2018 4:28:13 PM	20181203 162803	FTP

Markers

Memo

Version

Privileges

3. Execute Package - After updating materials in SAP table execute the package again.
4. Package is executed successfully and two records are updated in Oracle Target Database. Its record statistics is displayed as shown below:

The screenshot displays the Oracle APEX interface for a session named 'CDHDR_DEMO_OH2'. The 'Record Statistics' section is circled in green, showing the following data:

Statistic	Value
No. of Inserts	0
No. of Updates	2
No. of Deletes	0
No. of Errors	0
No. of Rows	8

5. Extracted Records - Below highlighted records are updated in the target database after package execution.

MANDT	MATNR	UDATE	TCODE	DELETED	MEINS
810	00000000000000000033	12/3/2018	MM02		KI
810	00000000000000000041	12/3/2018	MM02		PAK
810	00000000000000000039	11/19/2018	MM01		CCM
810	00000000000000000038	11/19/2018	MM01	X	G
810	00000000000000000037	11/19/2018	MM01	X	%
810	00000000000000000036	11/5/2018	MM01	X	KAR
810	00000000000000000035	11/5/2018	MM01	X	CCM
810	00000000000000000031	10/29/2018	MM01		%
810	00000000000000000034	10/29/2018	MM01		G
810	00000000000000000032	10/29/2018	MM01		GAU
810	00000000000000000023	7/30/2018	MM01		ST
810	00000000000000000022	7/26/2018	MM01		ST

Delete Records from SAP and Execute Package

For deleting records from SAP table:

1. From the below MADescripRA_REPLICATED Target table find the highlighted details before the flagging material for deletion.

MANDT	MATNR	UDATE	TCODE	DELETED	MEINS
810	00000000000000000033	12/3/2018	MM02		KI
810	00000000000000000041	12/3/2018	MM02		PAK
810	00000000000000000039	11/19/2018	MM01		CCM
810	00000000000000000038	11/19/2018	MM01	X	G
810	00000000000000000037	11/19/2018	MM01	X	%
810	00000000000000000036	11/5/2018	MM01	X	KAR
810	00000000000000000035	11/5/2018	MM01	X	CCM
810	00000000000000000031	10/29/2018	MM01	X	%
810	00000000000000000034	10/29/2018	MM01		G
810	00000000000000000032	10/29/2018	MM01		GAU
810	00000000000000000023	7/30/2018	MM01		ST
810	00000000000000000022	7/26/2018	MM01		ST

Material 31 and 34 are marked for deletion. Field LVORM (DF at client level) will be empty before deletion in the MARA Table which we mapped with field DELETED in Target table.

2. CDHDR table adds a new row for all the new material created, updated and flagged for deletion.

Data Browser: Table CDHDR Select Entries 6

Table: CDHDR
Displayed Fields: 14 of 14 Fixed Columns: [4] List Width 0

MANDANT	OBJECTCLAS	OBJECTID
<input type="checkbox"/> 810	MATERIAL	000000000000000000000031
<input type="checkbox"/> 810	MATERIAL	000000000000000000000034
<input type="checkbox"/> 810	MATERIAL	000000000000000000000041
<input type="checkbox"/> 810	MATERIAL	000000000000000000000041
<input type="checkbox"/> 810	MATERIAL	000000000000000000000041
<input type="checkbox"/> 810	MATERIAL	000000000000000000000041

TCode field holds the below values:

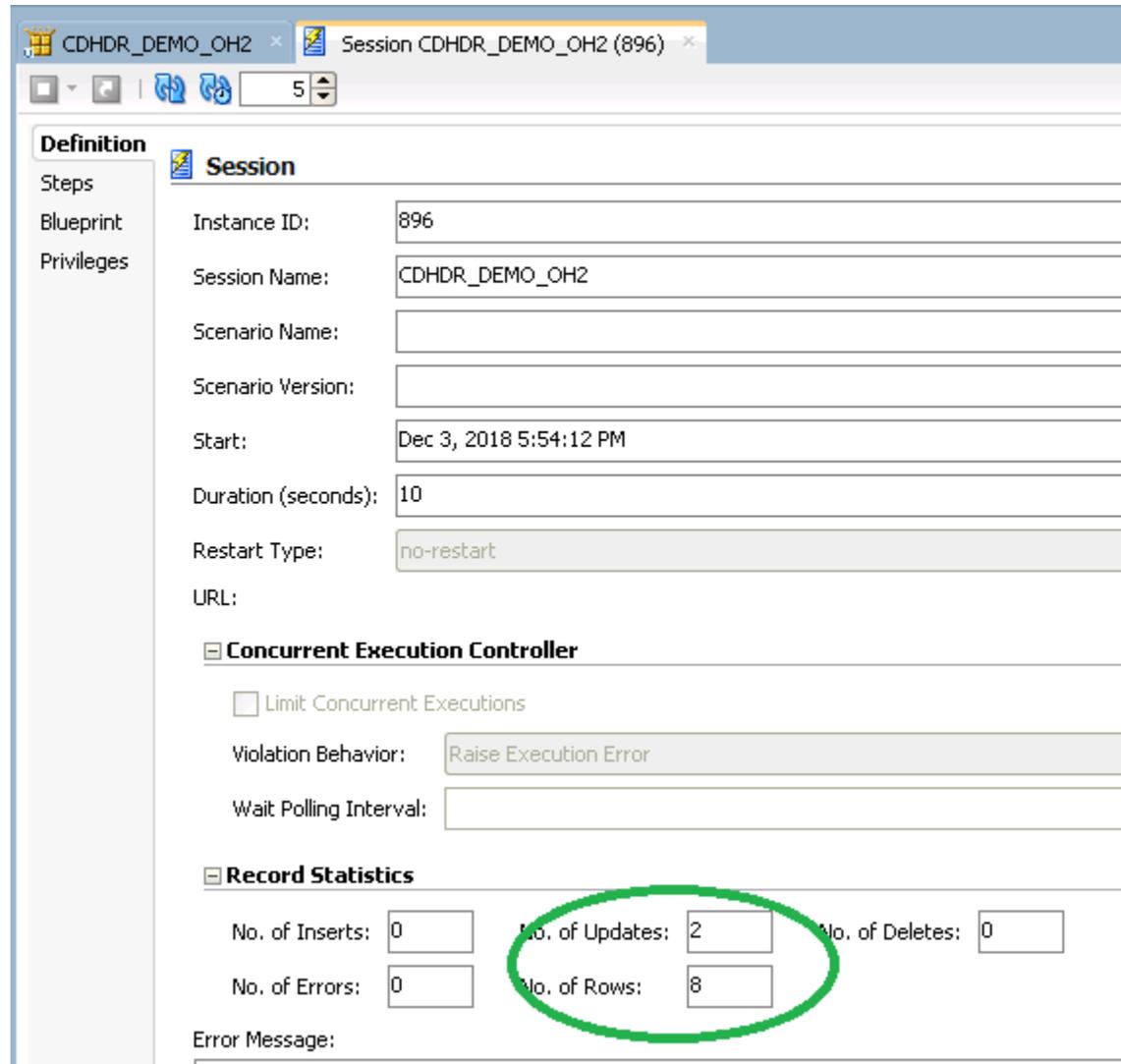
- a. MM01: New material created
 - b. MM02: Existing material updated
 - c. MM06: Material marked for deletion
3. MARA Table - If a material is marked for deletion a new row is added with Tcode : MM06 (Deletion). After flagging material for deletion Field LVORM (DF at client level) is updated with 'X'.

Data Browser: Table MARA Select Entries 3

Table: MARA
Displayed Fields: 22 of 237 Fixed Columns: [2] List Width 0

MANDT	MATNR	ERSDA	ERNAM	LAEDA	AENAM	VPSTA
<input type="checkbox"/> 810	000000000000000000000031	29.10.2018	I10670	03.12.2018	ZODI_RKMTEST	K
<input type="checkbox"/> 810	000000000000000000000033	29.10.2018	I10670	03.12.2018	ZODI_RKMTEST	K
<input type="checkbox"/> 810	000000000000000000000041	03.12.2018	I10670	03.12.2018	ZODI_RKMTEST	K

4. Execute package - Package execute success as find below instance there are two records are changed.



5. Extracted Records - After package execution find the extracted records in target oracle table MARA_REPLICATED as below.

SQL (recovered) * +

```

2
3 ▶ SELECT MANDT , MATNR , UDATE , TCODE , DELETED , MEINS FROM MARA_REPLICATED O

```

Data Grid

Messages | Data Grid | Trace | DBMS Output (disabled) | Query Viewer | Explain Plan | Script O

MANDT	MATNR	UDATE	TCODE	DELETED	MEINS
▶ 810	0000000000000000041	12/3/2018	MM02		PAK
810	0000000000000000034	12/3/2018	MM06	X	G
810	0000000000000000033	12/3/2018	MM02		KI
810	0000000000000000031	12/3/2018	MM06	X	%
810	0000000000000000039	11/19/2018	MM01		CCM
810	0000000000000000038	11/19/2018	MM01	X	G
810	0000000000000000037	11/19/2018	MM01	X	%
810	0000000000000000035	11/5/2018	MM01	X	CCM
810	0000000000000000036	11/5/2018	MM01	X	KAR
810	0000000000000000032	10/29/2018	MM01		GAU
810	0000000000000000023	7/30/2018	MM01		ST
810	0000000000000000022	7/26/2018	MM01		ST
810	0000000000000000021	7/18/2018	MM01		ST

3: 86 | 843 msec | Row 1 of 500 fetched so far (more rows exist) | DEV42REPO@10.30.32.71:1521/PDBORCL

CAPS | NUM | INS | Run to cursor

After t
Field D
will be
mappe

8

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 section 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* section 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](#).

A

SAP ABAP ERP Required Privileges

This appendix lists the required privileges for accessing the SAP system using Oracle Data Integrator.

An 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, an SAP RFC user is sufficient.

These SAP user types can be used for the following operations:

Table A-1 Required SAP User Types

Operation	Required SAP User Type
RKM: Setup	SAP Dialog user
RKM: Reverse Engineering	SAP Dialog user or SAP RFC user
LKM: Development	SAP Dialog user or SAP RFC user
LKM: Production	SAP Dialog user or SAP RFC user

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

- [Authorizations Required for RKM SAP ERP Upload](#)
- [Authorizations Required for RKM SAP ERP Execution](#)
- [Authorizations Required for LKM SAP ERP Upload](#)
- [Authorizations Required for LKM SAP ERP Execution](#)

Important points to consider

Consider the following points while configuring the SAP privileges:

- **S_DATASET** is an authorization object that controls access to physical file, so you need to provide access to SAP directories & folder mounted path. The values provided here are sample paths, you need to provide your landscape directories path or provide ***(all)**.
- **NR** or **(NR)** means **NOT REQUIRED**.

Authorizations Required for RKM SAP ERP Upload

The following SAP authorizations are required only in rare cases, when explicitly requested by Oracle product support. For standard installations, these authorizations are not required.

It is not required in TR-based installation.



Note:

Developer key is required in this authorization.

Object	Field Name	Value (ECC6-EHP3)	Value (ECC6-EHP7)
S_RFC	ACTVT	16	16
S_RFC	RFC_NAME	RFC1, SDIFRUNTIME, SUTL, SYST, ZODI_FGR_PROD*, ZODI_FGR	RFC1, SDIFRUNTIME, SUTL, SYST, ZODI_FGR_PROD*, ZODI_FGR
S_RFC	RFC_TYPE	FUGR	FUGR
S_TCODE	TCD	SU53,SE38	SE38, WE30, SU53
S_ADMI_FCD	S_ADMI_FCD	NR	RFCA
S_CTS_ADMI	CTS_ADMFCT	TABL	TABL
S_CTS_SADM	CTS_ADMFCT	NR	TABL
S_CTS_SADM	DESTSYS	NR	*
S_CTS_SADM	DOMAIN	NR	*
S_DATASET	ACTVT	34	33, 34
S_DATASET	FILENAME	*	*
S_DATASET	PROGRAM	SAPLSTRF	*
S_TABU_DIS	ACTVT	3	02, 03
S_TABU_DIS	DICBERCLS	*	*
S_TABU_NAM	ACTVT	NR	02, 03
S_TABU_NAM	TABLE	NR	*
S_DEVELOP	ACTVT	01,02,03	01, 02, 03, 06
S_DEVELOP	DEVCLAS	\$TMP, ZODI_LKM_PCKG, ZODI_RKM_PCKG	\$TMP, ZODI_LKM_PCKG, ZODI_RKM_PCKG
S_DEVELOP	OBJNAME	Z\$\$\$XRFC, ZODI*	Z\$\$\$XRFC, ZODI*
S_DEVELOP	OBJTYPE	DEVC, FUGR, FUNC, PROG	DEVC, FUGR, FUNC, PROG
S_DEVELOP	P_GROUP		' '
S_PROGRAM	P_ACTION	NR	BTCSSUBMIT, SUBMIT, VARIANT
S_PROGRAM	P_GROUP	NR	*

Object	Field Name	Value (ECC6-EHP3)	Value (ECC6-EHP7)
S_SYS_RWBO	ACTVT	NR	01, 02, 03, 05, 06, 23
S_SYS_RWBO	DESTSYS	NR	*
S_SYS_RWBO	DOMAIN	NR	' '
S_SYS_RWBO	TTYPE	NR	DTRA
S_TRANSPRT	ACTVT	01,03	01, 02, 03, 06
S_TRANSPRT	TTYPE	DTRA, TASK	*
S_IDOCDEFT	ACTVT	NR	3
S_IDOCDEFT	EDI_CIM	NR	
S_IDOCDEFT	EDI_DOC	NR	
S_IDOCDEFT	EDI_TCD	NR	

Authorizations Required for RKM SAP ERP Execution

The following SAP authorizations are required for reverse engineering the SAP models using **RKM SAP ERP**.

Object	Field Name	Value (ECC6-EHP3)	Value (ECC6-EHP7)
S_RFC	ACTVT	16	16
S_RFC	RFC_NAME	RFC1, SDIFRUNTIME, SUTL, SYST, ZODI_FGR_PROD*, ZODI_FGR	RFC1, SDIFRUNTIME, SUTL, SYST, ZODI_FGR_PROD*, ZODI_FGR
S_RFC	RFC_TYPE	FUGR	FUGR
S_TCODE	TCD	SU53, SE38	SU53, SE38
S_CTS_ADMI	CTS_ADMFCT	NR	TABL
S_DATASET	ACTVT	NR	34
S_DATASET	FILENAME	NR	*
S_DATASET	PROGRAM	NR	SAPLSTRF
S_TABU_DIS	ACTVT	03	02, 03
S_TABU_DIS	DICBERCLS	*	*
S_TABU_NAM	ACTVT	NR	02, 03
S_TABU_NAM	TABLE	NR	*
S_TRANSPRT	ACTVT	01, 03	01, 03
S_TRANSPRT	TTYPE	DTRA, TASK	DTRA, TASK

Authorizations Required for LKM SAP ERP Upload

The following SAP authorizations are required for running mappings using the SAP ERP LKMs in SAP development environment with Upload ABAP Option Enabled.



Note:

Developer Key is required in this authorization.

Object	Field Name	Value (ECC6-EHP3)	Value (ECC6-EHP7)
S_RFC	ACTVT	16	16
S_RFC	RFC_NAME	RFC1, SDIFRUNTIME, SYST, ZODI_FGR, ZODI_FGR_PROD_*	RFC1, SDIFRUNTIME, SYST, ZODI_FGR, ZODI_FGR_PROD_*
S_RFC	RFC_TYPE	FUGR	FUGR
S_TCODE	TCD	SU53, SE38	SE38, SU53
S_ADMI_FCD	S_ADMI_FCD	RFCA, SFTP	RFCA, SFTP
S_BTCH_ADM	BTCADMIN	Y	Y
S_BTCH_JOB	JOBACTION	RELE	RELE
S_BTCH_JOB	JOBGROUP	' '	' '
S_CTS_ADMI	CTS_ADMFCT	TABL	TABL
S_DATASET	ACTVT	06,33,34	06,33,34
S_DATASET	FILENAME	*	*
S_DATASET	PROGRAM	*	SAPLSLOG, SAPLSTRF, SAPLZ*, ZODI*
S_DEVELOP	ACTVT	01	01
S_DEVELOP	DEVCLASS	*	*
S_DEVELOP	OBJNAME	ZODI*	ZODI*
S_DEVELOP	OBJTYPE	FUGR, FUNC	FUGR, FUNC
S_DEVELOP	P_GROUP	NR	
S_TRANSPRT	ACTVT	01, 02	01, 02
S_TRANSPRT	TTYPE	DTRA, TASK	DTRA, TASK
S_APPL_LOG	ACTVT	06	6
S_APPL_LOG	ALG_OBJECT	ZODI_APPOBJ	ZODI_APPOBJ
S_APPL_LOG	ALG_SUBOBJ	ZODI_SUBOBJ	ZODI_SUBOBJ
S_TABU_DIS	ACTVT	3	3
S_TABU_DIS	DICBERCLS	*	*

Authorizations Required for LKM SAP ERP Execution

The following SAP authorizations are required for running mappings using the SAP ERP LKMs in a SAP production environment, when LKM option Upload ABAP Code is disabled (QA or Production).

Object	Field Name	Value (ECC6-EHP3)	Value (ECC6-EHP7)
S_RFC	ACTVT	16	16
S_RFC	RFC_NAME	RFC1, SDIFRUNTIME, SYST, ZODI_FGR_PROD*	RFC1, SDIFRUNTIME, SYST, ZODI_FGR_PROD*
S_RFC	RFC_TYPE	FUGR	FUGR
S_ADMI_FCD	S_ADMI_FCD	SFTP	SFTP
S_BTCH_ADM	BTCADMIN	Y	Y
S_BTCH_JOB	JOBACTION	RELE	RELE
S_BTCH_JOB	JOBGROUP	' '	' '
S_DATASET	ACTVT	06,33,34	06,33,34
S_DATASET	FILENAME	*	*
S_DATASET	PROGRAM	SAPLSLOG*, SAPLSTRF*, SAPLZODI*, ZODI*, SAPLZODI_FGR_PROD*	SAPLSLOG*, SAPLSTRF*, SAPLZODI*, SAPLZODI_FGR_PROD*, ZODI*
S_TRANSPRT	ACTVT	01, 02	01, 02
S_TRANSPRT	TTYPE	DTRA, TASK	DTRA, TASK
S_APPL_LOG	ACTVT	06	06
S_APPL_LOG	ALG_OBJECT	ZODI_APPOBJ	ZODI_APPOBJ
S_APPL_LOG	ALG_SUBOBJ	ZODI_SUBOBJ	ZODI_SUBOBJ

B

SAP Stand-Alone Connection Test

This appendix describes how to perform an 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.

SAP Stand-Alone Connection Test

To use a standalone java utility to test the connection:

1. Open a command window.
2. Change to <ODI_HOME>/odi/sdk/lib 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 <ODI_HOME>/odi/sdk/lib directory.
5. Launch the utility using the following command:

On Windows:

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

On Linux/UNIX:

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

This should result in an output similar to this:

```
JCO3 Library available!!
JCO Version: 3.0.2 (2007-03-29)
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 oraledi/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 your 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:

On Windows:

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

On Linux/UNIX:

```
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 Library 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:
SYSNR: 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_CODE and UPLOAD_ABAP_BASE set to true).

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

C

Uninstalling ODI SAP Components

This appendix describes how to uninstall ODI components from your SAP system. For removing ODI connectivity from the SAP system, please follow the steps described in this section.

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

Updating ODI SAP Components

During first-time installation the TR installs some ODI objects into the SAP system. This installation consists of two parts: some base objects and some RFCs. Such full installation requires that no ODI SAP objects are installed in the SAP system. See [Uninstalling ODI SAP Components](#) if needed.

If the ODI objects installed into your SAP systems have been installed by RKM SAP ERP v34 or later / RKM SAP BW v25 or later, there is no need for reinstalling the base objects and it is sufficient to update just the RFCs. Reinstalling the RFCs is achieved by executing a reverse engineering run with RKM option `UPLOAD_ABAP_BASE` set to `false` and RKM option `UPLOAD_ABAP_CODE` set to `true`.

Uninstalling ODI SAP Components

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_RKM_PCKG` in the package or development class field. Note that for older installations the package name is `ZODI_DEVCLASS`.
6. Right-click the `ZODI_RKM_PCKG` object below the object name tab.
7. Select **Delete** in the drop down menu and delete the development class.

If it does not allow deleting the development class individually, delete all the objects inside one by one. For the contained Function Group please make sure to delete it in one go and do not delete the Function Group's Function Modules individually. Once all objects have been removed, delete the development class itself.

8. Go to transaction `SE10`. Select **Modifiable** and **Workbench Requests**. Click **Display**.
9. Select the transport requests which have the ODI Objects and Release them.
10. Now repeat the steps 3 to 7 to delete Development Class.

11. Go to transaction SE01. Select **Modifiable** and **Workbench Requests**. Click **Display**. Release the Transport Request which has ZODI_RKM_PCKG.
12. Repeat Step 4 to 11 for package ZODI_LKM_PCKG.