

Oracle® Fusion Middleware

Getting Started with SAP ABAP BW Adapter for Oracle Data
Integrator

12c (12.1.2)

E41954-02

December 2013

Oracle Fusion Middleware Getting Started with SAP ABAP BW Adapter for Oracle Data Integrator, 12c (12.1.2)

E41954-02

Copyright © 2010, 2013, Oracle and/or its affiliates. All rights reserved.

Primary Authors: Laura Hofman, Joshua Stanley

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, 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 failsafe, 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 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.

Contents

Preface	vii
Audience	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions	viii
1 Overview of Oracle Data Integrator SAP BW Knowledge Modules	
Overview	1-1
2 Setting Up the Environment	
Before You Begin	2-1
System Requirements and Certification	2-1
Getting the Right Privileges	2-2
Gathering SAP Connection Information	2-2
Validating the Shared Folder Setup	2-3
Validating the FTP Setup	2-4
Validating SAP Privileges	2-5
Validating SAP Transport Layer Name	2-5
Installing Oracle Data Integrator	2-6
Installing and Configuring the Oracle DB Target	2-6
Installing and Configuring JCo	2-9
Setting Up an FTP Server	2-10
Configuring Oracle Data Integrator	2-10
Set Up the Topology	2-10
Add the Open Tool	2-11
3 Configuring the Topology	
Configuring the Target Data Server	3-2
Configuring the SAP Source Server	3-3
Configuring the Data Server	3-3
Configuring the Logical Schema	3-4
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 BW Source Datastores	5-2
Creating the Data Model	5-3
Validating the SAP Connection	5-3
Starting the Reverse-Engineering Process	5-4
Using the SAP Metadata Browser	5-5
The Tree View	5-6
The Search Panel	5-7
Reverse-Engineering Without GUI	5-8

6 Creating and Running the Mapping

Creating the Mapping	6-1
Info Cube Extraction	6-2
Create the Mapping	6-2
Define the Source and Target Datastores	6-2
Create the Mappings	6-2
Define the Mapping Flow	6-3
Info Object Extraction	6-4
Create the Mapping	6-4
Define the Source and Target Datastores	6-4
Define Joins between Info Object's Main table and Text Table	6-4
Create the Mappings	6-6
Define the Mapping Flow	6-6
Info Object Hierarchy Extraction	6-7
Create the Mapping	6-7
Define the Source and Target Datastores	6-7
Create the Mappings	6-8
Define the Mapping Flow	6-8
Info Object Interval Hierarchy Extraction	6-9
Create the Mapping	6-9
Define the Source and Target Datastores	6-9
Define Joins between Sources (Info Object Main table and InfoObject Hierarchy Table)	6-10
Create the Mappings	6-11
Define the Mapping Flow	6-12
ODS/DSO Extraction	6-13
Create the Mapping	6-13
Define the Source and Target Datastores	6-13
Create the Mappings	6-13
Define the Mapping Flow	6-14
Info Cube & Related Info Object Extraction	6-14
Create the Mapping	6-14
Define the Source and Target Datastores	6-15
Define Joins between Sources (Info Cube & Object)	6-15
Create the Mappings	6-16
Define the Mapping Flow	6-17

Info Cube Extraction with Navigational Attributes	6-17
Create the Mapping	6-18
Define the Source and Target Datastores	6-18
Define Joins Between Sources (Info Cube & Info Object)	6-18
Create the Mappings	6-19
Define the Mapping Flow	6-20
Open Hub Extraction	6-21
Create the Mapping	6-21
Define the Source and Target Datastores	6-21
Create the Mappings	6-22
Define the Mapping Flow	6-22
Running the Mapping	6-23
Running the Mapping	6-23
Review the Mapping Execution	6-24
Review the Resulting Data	6-24
7 Creating a Package for Delta Extraction	
Modify the Mapping Flow	7-1
Creating the MIN_REQUEST_ID Variable	7-2
Creating a Package for Delta Extraction	7-3
Running the Package	7-4
8 Going Further with Oracle Data Integrator	
Going further with Oracle Data Integrator	8-1
A SAP ABAP BW Required Privileges	
Important points to consider	A-1
Authorizations Required for RKM SAP BW Upload	A-2
Authorizations Required for RKM SAP BW Execution	A-3
Authorizations Required for LKM SAP BW Upload (Background mode)	A-4
Authorizations Required for LKM SAP BW Execution (Background mode)	A-5
Authorizations Required for LKM SAP BW Upload (Dialog mode)	A-5
Authorizations Required for LKM SAP BW Execution (Dialog mode)	A-6
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 BW Knowledge Modules and how to use them in Oracle Data Integrator.

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

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

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

It is recommended that you also review the SAP ABAP - BW chapter of the *Knowledge Module Developer's Guide for Oracle Data Integrator* for more information about the SAP BW KMs.

Audience

This document is intended for developers and administrators who want to use Oracle Data Integrator Application Adapters with SAP BW 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 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 Oracle resources:

- *Oracle Fusion Middleware Getting Started with Oracle Data Integrator*
- *Application Adapters Guide for Oracle Data Integrator*
- *Installing and Configuring Oracle Data Integrator*
- *Developer's Guide for Oracle Data Integrator*
- *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*
- *Knowledge Module Developer's Guide for Oracle Data Integrator*
- *Getting Started with SAP ABAP ERP Adapter for Oracle Data Integrator*
- *Oracle Data Integrator Studio Help*
- *Release Notes for Oracle Data Integrator*

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.

Overview of Oracle Data Integrator SAP BW Knowledge Modules

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

Overview

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

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

This adapter includes two knowledge modules:

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

Setting Up the Environment

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

This chapter includes the following sections:

- [Before You Begin](#)
- [Validating the Shared Folder Setup](#)
- [Validating the FTP Setup](#)
- [Validating SAP Privileges](#)
- [Validating SAP Transport Layer Name](#)
- [Installing Oracle Data Integrator](#)
- [Installing and Configuring the Oracle DB Target](#)
- [Installing and Configuring JCo](#)
- [Setting Up an FTP Server](#)
- [Configuring Oracle Data Integrator](#)

Before You Begin

Before you begin, consider the following:

- [System Requirements and Certification](#)
- [Getting the Right Privileges](#)
- [Gathering SAP Connection Information](#)

System Requirements and Certification

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

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

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

- Oracle Data Integrator 11.1.1.7.0 or above.

- A Java Connector (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 Section 8.7.1, "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 BW 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 BW to Oracle (SQLLDR) requires a FTP server to upload data from the SAP BW 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 BW machine and the agent machine.
- For LKM SAP BW to Oracle (SQLLDR) only: SQLLoader is required on the machine running the agent. SQLLoader is used for loading data extracted from SAP BW system 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 BW Required Privileges."](#)

These privileges are required for the SAP BW user which will be used to connect SAP BW system by the ODI SAP BW Adapter

Please note that a developer key is mandatory for the SAP user.

Gathering SAP Connection Information

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

- **SAP BW 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 BW 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 BW Version:** BW 3.5 or BI 7.0.
- **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 **Testconnection**.

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 BW 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 *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 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 tables
-- For ODS/DSO

CREATE TABLE ODI_SAP_DEMO.W_PURCH_DS (
  PO_GUID VARCHAR2(100),
  ORDER_ITEM_GUID VARCHAR2(100),
  ACC_ASS_GUID VARCHAR2(100),
  GUIDE_CONTRCT VARCHAR2(100),
  ORDER_QUANT NUMBER,
  NET_PRICE NUMBER,
  PRICE_UNIT CHAR(5),
```



```
NO_OF_PURCH_ODR NUMBER);
ALTER TABLE ODI_SAP_DEMO.W_PURCH_DS ADD CONSTRAINT PK_W_PURCH_DS PRIMARY KEY (PO_
GUID);

-- For INFO OBJECT

CREATE TABLE ODI_SAP_DEMO.W_ACCNT_VENDR_DS (
VENDOR CHAR(10),
PCOMPANY CHAR(10),
PHONE CHAR(20),
PLANT CHAR(10),
STREET VARCHAR(50),
MED_DESC VARCHAR(100));
ALTER TABLE ODI_SAP_DEMO.W_ACCNT_VENDR_DS ADD CONSTRAINT PK_W_ACCNT_VENDR_DS
PRIMARY KEY (VENDOR);

-- For INFO OBJECT HIERARCHY

CREATE TABLE ODI_SAP_DEMO.W_GL_ACCOUNT_HIER_DS (
HIER_NAME CHAR(50),
NODEID NUMBER(20),
INFOBJECT_NAME CHAR(80),
NODENAME CHAR(80),
TLEVEL NUMBER(5),
PARENTID NUMBER(20),
CHILDID NUMBER(20),
NEXTID NUMBER(20));

ALTER TABLE ODI_SAP_DEMO.W_GL_ACCOUNT_HIER_DS ADD CONSTRAINT PK_W_GL_ACCOUNT_HIER_
DS PRIMARY KEY (HIER_NAME, NODEID);

-- For INFO OBJECT INTERVAL HIERARCHY

CREATE TABLE TARGET_INTERVAL_HIERARCHY
(
RSHIENM VARCHAR2(30 BYTE),
NODEID NUMBER(8),
PARENTID NUMBER(8)
INTERVL VARCHAR2(1 BYTE)
LEAFFROM VARCHAR2(60 BYTE),
LEAFTO VARCHAR2(60 BYTE),
VERSION VARCHAR2(3 BYTE),
CHRT_ACCTS VARCHAR2(4 BYTE),
GL_ACCOUNT VARCHAR2(10 BYTE),
BAL_FLAG VARCHAR2(1 BYTE),
GLACCEXT VARCHAR2(18 BYTE),
INCST_FLAG VARCHAR2(2 BYTE)
)

-- For INFO CUBE

CREATE TABLE ODI_SAP_DEMO.W_VENDR_BAL_DS (
OCOMP_CODE CHAR(6),
OCREDITOR CHAR(10),
ORC_ACCOUNT CHAR(10),
TOT_DEBT_POST NUMBER,
TOT_CR_POST NUMBER,
```

```

CUMULATIVE_BAL NUMBER);

ALTER TABLE ODI_SAP_DEMO.W_VENDR_BAL_DS ADD CONSTRAINT PK_W_VENDR_BAL_DS PRIMARY
KEY (OCOMP_CODE);

-- For INFO CUBE & INFO OBJECT Join

CREATE TABLE ODI_SAP_DEMO.W_VENDER_CUBE_OBJ_DS (
COMP_CODE CHAR(10),
VENDOR CHAR(10),
GL_ACCOUNT CHAR(10),
SALES_FOR_PERIOD NUMBER,
CUM_BALANCE NUMBER,
PLANT CHAR(10),
MEDIUM_DESC VARCHAR2(100));

ALTER TABLE ODI_SAP_DEMO.W_VENDER_CUBE_OBJ_DS ADD CONSTRAINT PK_W_VENDER_CUBE_OBJ_
DS PRIMARY KEY (COMP_CODE);

CREATE TABLE "TARGET_NAVIGATIONAL_ATTRIBUTES"
(
IOPK_CUSTOMER VARCHAR2(10 BYTE),
IOPKCUST_GROUPVARCHAR2(2 BYTE),
IOPKPK_CUSTOMERVARCHAR2(10 BYTE),
IOPK_PK_PLANT VARCHAR2(4 BYTE),
IOPKCALMONTH NUMBER(6),
IOPK_CALWEEK NUMBER(6),
IOPKTHIP_POINTVARCHAR2(4 BYTE),
REQUIDVARCHAR2(30 BYTE),
KF_ODEL_VOL NUMBER(17,3),
KF_OSHIPDATE NUMBER(17),
KF_OLOADDATEK NUMBER(17),
KF_OACT_GI_DTKNUMBER(17),
KF_OMATAV_DATKNUMBER(17),
KF_OLATE_QTY NUMBER(17,3),
CUSTOMERVARCHAR2(10 BYTE),
PLANT VARCHAR2(4 BYTE),
PHONE VARCHAR2(16 BYTE),
PCOMPANYVARCHAR2(6 BYTE),
OUTL_TYPEVARCHAR2(4 BYTE),
NIELSEN_ID VARCHAR2(2 BYTE),
LOGSYSVARCHAR2(10 BYTE),
VISIT_RYTVARCHAR2(4 BYTE),
USAGE_INDVARCHAR2(3 BYTE),
SRCID VARCHAR2(4 BYTE),
PRECISIDNUMBER(4),
POSTCD_BOX VARCHAR2(10 BYTE),
CUS_F_CONS VARCHAR2(1 BYTE),
COUNTRYVARCHAR2(3 BYTE),
BPARTNERVARCHAR2(10 BYTE),
ACCNT_GRPVARCHAR2(4 BYTE),
LANGU VARCHAR2(1 BYTE),
KEYACCOUNT VARCHAR2(1 BYTE)
)

-- For OPEN HUB DESTINATION

CREATE TABLE ODI_SAP_DEMO.W_OPENHUB_SALES_DS (
OHREQUID NUMBER,

```

```

DATA_PACKAGE NUMBER,
DATA_RECORD NUMBER,
CUSTOMER CHAR(1),
MATERIAL CHAR(1),
MATERIAL_PRICE NUMBER,
CURRENCY NUMBER,
QUANTITY NUMBER,
TOTAL_REVENUE NUMBER);

```

```

ALTER TABLE ODI_SAP_DEMO.W_OPENHUB_SALES_DS ADD CONSTRAINT PK_W_OPENHUB_SALES_DS
PRIMARY KEY (OHREQUID);

```

Installing and Configuring JCo

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

To install and configure JCo:

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

Notes:

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

2. Unzip the appropriate distribution package into an arbitrary directory `<sapjco-install-path>`.
3. Follow the installation instructions in `<sapjco-install-path>/javadoc/installation.html` for the respective platform.
4. Copy `sapjco3.jar` and `sapjco3.dll`:
 - For ODI 10g: Copy `sapjco3.jar` and `sapjco3.dll` (or respective binary like `libsapjco3.so`) into the `oracledi/drivers` directory.
 - For ODI 11g: Copy `sapjco3.jar` and `sapjco3.dll` (or respective binary like `libsapjco3.so`) into the `<ODI_HOME>/odi_misc` directory (ODI Studio) and into the `<ODI_HOME>/oracledi/agent/drivers` directory (Standalone Agent).
 - For ODI 12c: Copy `sapjco3.jar` and `sapjco3.dll` (or respective binary like `libsapjco3.so`) into the `<ODI_HOME>/odi/sdk/lib` directory.
5. Restart ODI Studio and any agents.
6. Check the JCo installation. This will be also checked later in this guide.

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

Perform the following steps to configure Oracle Data Integrator:

1. [Set Up the Topology](#)
2. [Add the Open Tool](#)

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 *Installing and Configuring Oracle Data Integrator* for more information on the Master Repository upgrade process.
4. In Topology Manager, open the JavaBeanShell technology and check on the Language tab that the JYTHON language is listed. If not, add it.
5. Create a File data server pointing to an existing FTP 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. For use with "LKM SAP BW 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 section 3.3.1, "Creating a File Data Server" of *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*. The encoding chosen on this URL must match the code page used by the SAP Application Server.
7. In this File data server create a Physical Schema representing the folder in the 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 BW 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 BW to Oracle (SQLLDR)/ LKM SAP BW to SQL is set to NONE. As Work Schema is an OS file name, slashes/backslashes should be used according to OS.

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`.

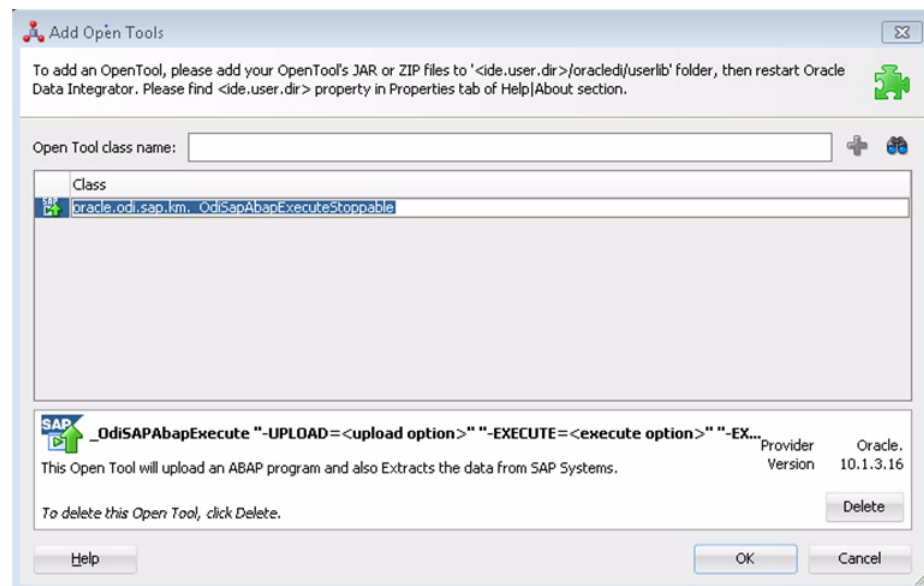
Refer to Section 7.7.1, "File Transfer Considerations" of *Application Adapters Guide for Oracle Data Integrator*.

- 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 Appendix A, "Oracle Data Integrator Tools Reference" of *Developer's Guide for Oracle Data Integrator*. This section details only the SAP ABAP specific steps.

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



- Click **OK**.

Configuring the Topology

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

This project demonstrates how to extract data from the SAP BW data targets listed below:

- **ODS/DSO:**

Source data target *0BBP_PO* from SAP BW is the ODS containing vendor information.

Target *W_PURCH_DS* is a table in an Oracle database that needs to be loaded from the SAP BW Sources. This table contains purchase order info.

- **Info Object:**

Source data target *0CREDITOR* is the Info object from SAP BW containing Account number of supplier/vendor.

Target *W_ACCNT_VENDR_DS* is a table in an Oracle database that needs to be loaded from the SAP BW Sources. This table contains account number info of supplier/vendor.

- **Info Object Hierarchy:**

Source data target *0GL_ACCOUNT* is the Info object from SAP BW containing General Ledger Information.

Target *W_GL_ACCOUNT_HIER_DS* is a table in an Oracle database that needs to be loaded from the SAP BW Sources. This table contains account number info of supplier/vendor.

- **Info Cube:**

Source data target *0FIAP_C20* is the Info cube from SAP BW containing vendor balances.

Target *W_VENDR_BAL_DS* is a table in an Oracle database that needs to be loaded from the SAP BW Sources. This table contains vendor balances.

- **Info Cube & related Info Objects:**

Source data targets *0FIAP_C20* and *0CREDITOR* are the Info Cube & Info Object respectively both containing the vendor related information.

Target *W_VENDER_CUBE_OBJ_DS* is a table in an Oracle database that needs to be loaded from the SAP BW Sources as per join condition.

- **Open Hub:**

Source data target *ZOHD_SALE* is an Open Hub destination name and its associated table is */BIC/OHZOHD_SALE* which contains the sales information.

Target *W_OPENHUB_SALES_DS* is a table in an Oracle database that needs to be loaded from the SAP BW Sources. This table contains sales information.

- **Delta Extraction:**

For delta Extraction you need to pass the value for two KM options i.e. *FIRST_REQ_ID* and *LAST_REQ_ID* from any above said data targets except Info Object after full load. The *FIRST_REQ_ID* is the start point of Delta Extraction and *LAST_REQ_ID* is the end point of Delta Extraction.

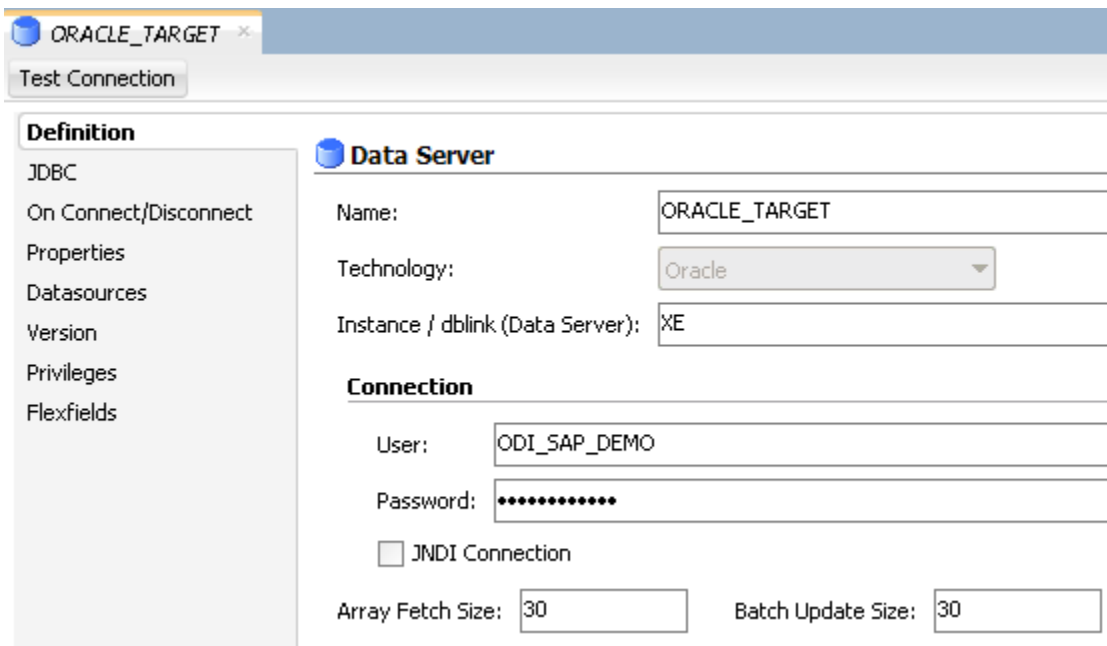
This chapter includes the following sections:

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

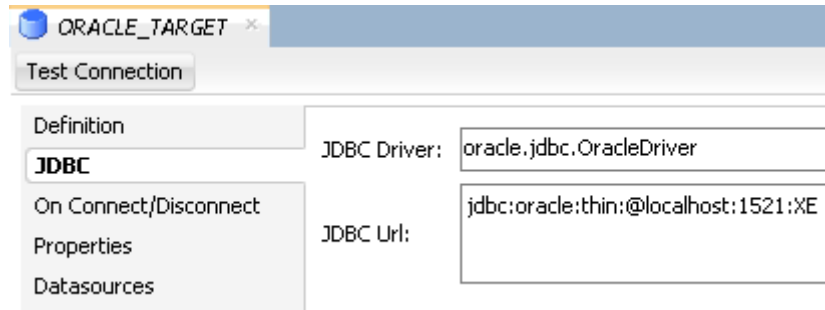
Configuring the Target Data Server

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 BW 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 the **File** menu, click **Save**.
8. Right-click the data server you created and select **New Physical Schema**.
9. In the Physical Schema windows that appears, select the ODI_SAP_DEMO for both Schema and Work Schema.

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

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

Configuring the SAP Source Server

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

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

Configuring the Data Server

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_BW. The name of the data server as it will appear in ODI.
 - **Host (Data Server):** SAP BW 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 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.

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 "[Set Up the Topology](#)" on page 2-10 for non-UNICODE SAP systems.

- **SAP BW Version:** Enter the SAP BW Version as follows:
 For SAP BW 7.0 and higher systems enter 700
 For SAP BI 3.5 systems enter 350
 - **SAP ERP Version & SAP ABAP Version:** unused in SAP BW Connector.
5. From the **File** menu, click **Save**.

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

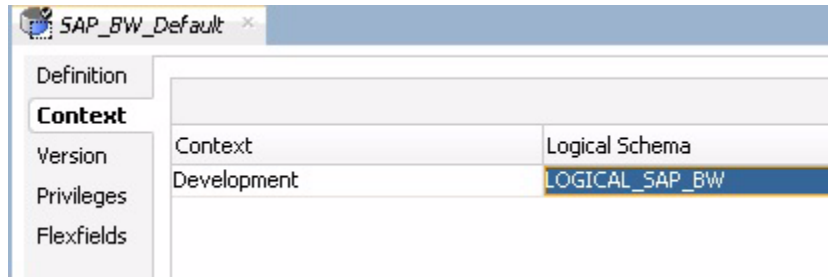
Except for the data server name, all the parameters that you provide while defining the SAP Data server should be provided by the SAP Administrators. See "[Getting the Right Privileges](#)" on page 2-2 for more information.

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 BW 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](#)" on page 2-2 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_BW.



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

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 BW:

1. Connect to Designer.
2. In the Projects tree view click **New Project**.
3. In the Definition tab, enter `SAP BW 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_BW_DataTargets`.
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 BW KMs. By default, they are located in the `../impexp` directory.
10. Select the following KMs:
 - IKM Oracle Incremental Update
 - LKM SAP BW to Oracle (SQLLDR)
 - LKM SAP BW to SQL
 - RKM SAP BW
 - RKM SAP ERP Connection Test
11. Click **OK** to perform the import.

Reverse-Engineering Data Models

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

This chapter includes the following sections:

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

Reverse-Engineering the Oracle Target

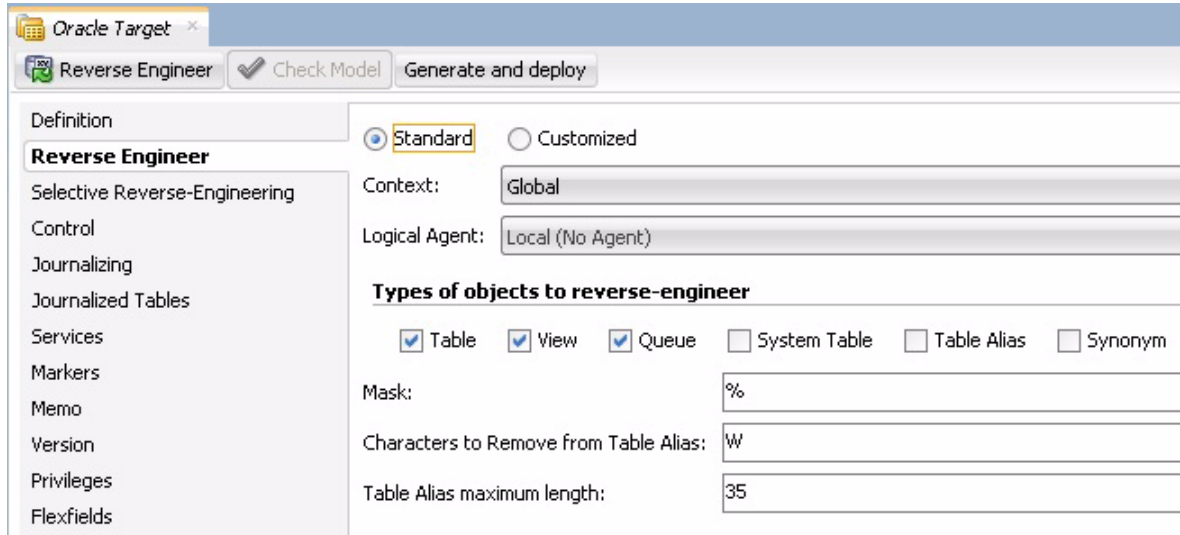
To reverse-engineer the Oracle data model:

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

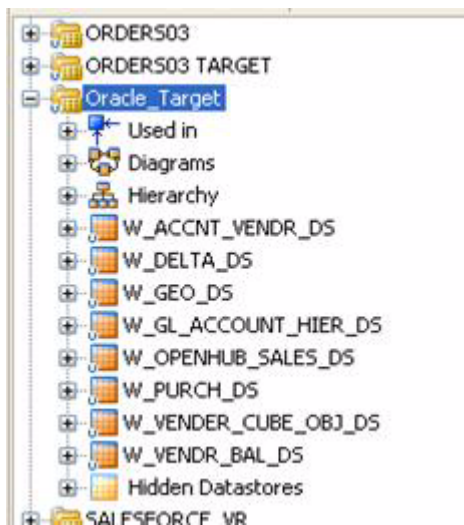
The screenshot shows the 'Definition' tab in the ODI interface. On the left, there is a 'Definition' sidebar with options like Reverse Engineer, Control, Journalizing, etc. The main area is titled 'Model' and contains several input fields:

- 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 Engineer tab, select the **Global** context.



5. Click **Reverse Engineer**, then from the **File** menu, click **Save**.
6. The W_PURCH_DS, W_ACCNT_VENDR_DS, W_GL_ACCOUNT_HIER_DS, W_VENDR_BAL_DS, W_VENDER_CUBE_OBJ_DS, W_OPENHUB_SALES_DS, W_DELTA_DS datastore appear in your Oracle_Target model, as shown below.



Reverse-Engineering the SAP BW Source Datastores

To reverse-engineer the SAP BW Source data targets you need to perform the following tasks:

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

This section also includes the following topics:

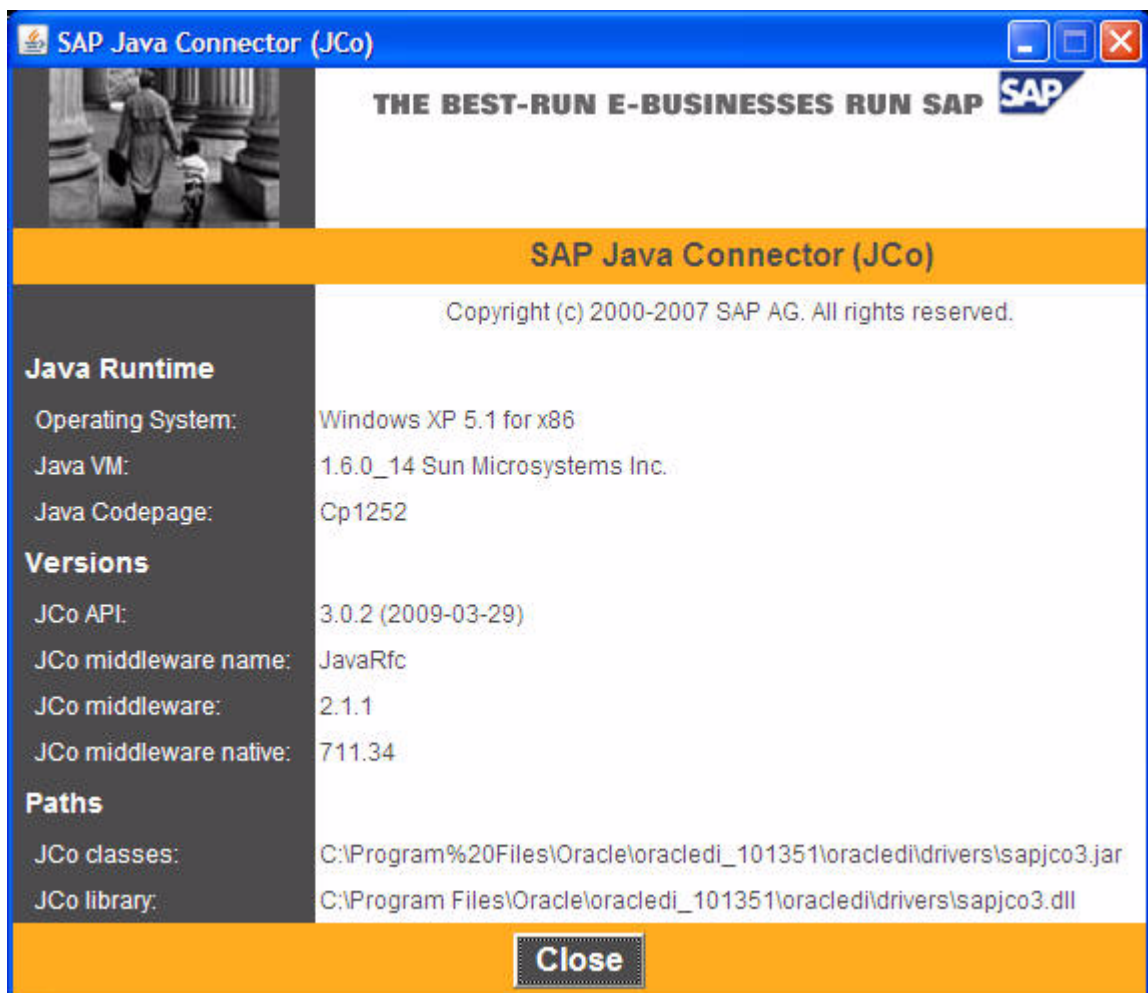
- [Using the SAP Metadata Browser](#)
- [Reverse-Engineering Without GUI](#)

Creating the Data Model

1. In the Models tree view, click **New Model**.
2. In the Definition tab, select SAP ABAP technology and the Logical Schema previously created (in our example it is the LOGICAL_SAP_BW Logical Schema).
3. In the Reverse Engineer tab:
 1. Select Customized option.
 2. Select the Global context.
 3. Select the RKM SAP Connection Test you have imported in the SAP BW Demo project.

Validating the SAP Connection

1. In the Models tree view, open the SAP BW 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 BW system and
 - Validate the proper setup of SAP JCo by displaying the About dialog.
3. In a few seconds, you should see the SAP JCo-About dialog as shown here:



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

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

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

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

Note: If you want to validate the SAP connectivity independent of ODI, please see [Appendix B, "SAP Stand-Alone Connection Test."](#)

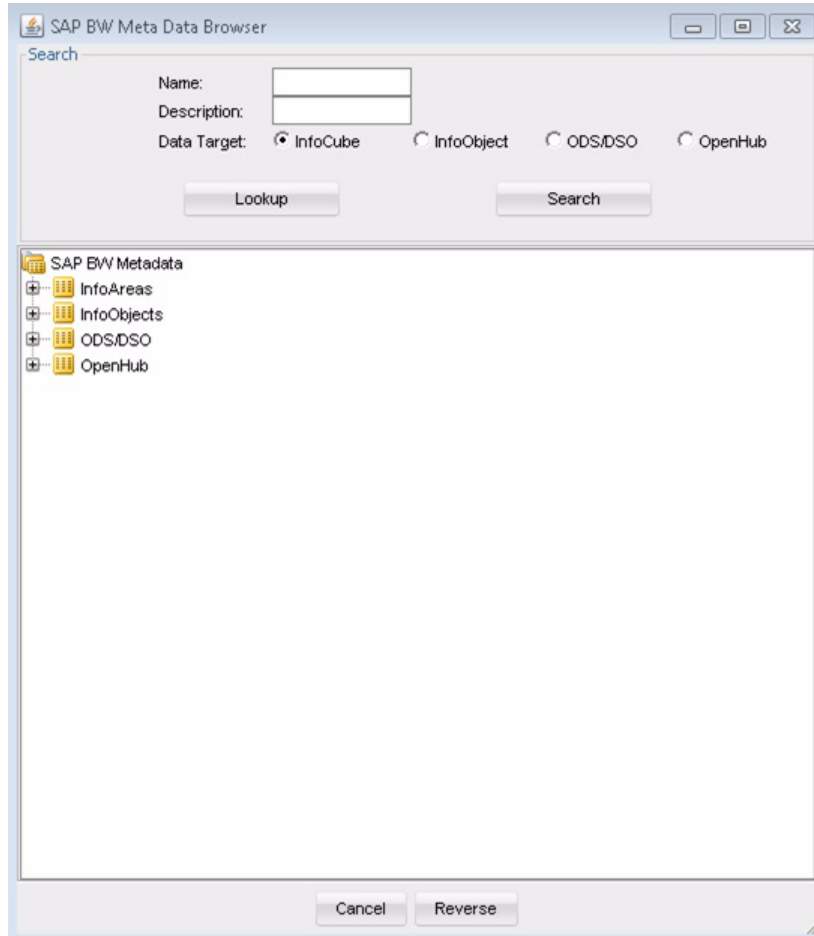
Starting the Reverse-Engineering Process

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

1. In the Models tree view, open the SAP BW Source Model.
2. In the Reverse Engineer tab:
 - a. Select the Customized option.
 - b. Select the Global context.
 - c. Select the RKM SAP BW you have imported in the SAP BW 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:

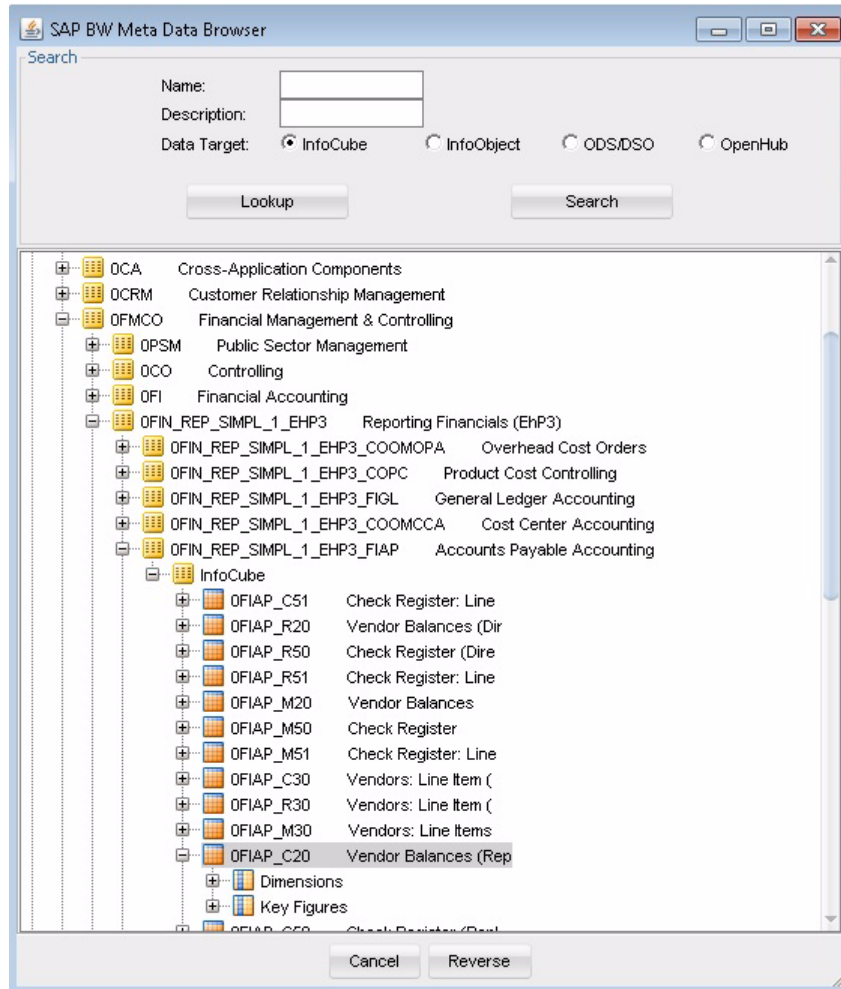
- Pay particular attention to provide the correct value for the SAP_TRANSPORT_LAYER_NAME option. If an invalid value is used during the first time installation, a significant delay in the connector installation may be caused.
 - For first time RKM use only: If you are using a custom value for KM options SAP_FUNCTION_GROUP_ODI_DEV and SAP_FUNCTION_GROUP_ODI_PROD, these function groups must *not* exist prior the first execution of the UPLOAD_ABAP_CODE and UPLOAD_ABAP_BASE set to true.
 - For first time RKM use only: If you are using a custom value for KM options SAP_PACKAGE_NAME_ODI_DEV and SAP_PACKAGE_NAME_ODI_PROD, these packages must *not* exist prior the first execution of the UPLOAD_ABAP_CODE and UPLOAD_ABAP_BASE set to true.
-
-



The Tree View

The SAP BW Data targets are organized into a tree view. Only the SAP BW Data targets available in the SAP BW system are displayed. By expanding the tree view, you can see Info Cubes, Info Objects, ODS/DSO, Open Hub, and select those that you want to reverse-engineer.

Note that only the data targets 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:

- Info Cube Name
- Info Object Name
- ODS/DSO
- Open Hub

For the example project for Info Cube, the data target 0FIAP_C20 is used:

1. Enter 0FIAP_C20 in the Name field.
2. Select the **InfoCube** data target.
3. Click **Search** to find the object to reverse-engineer. The results are highlighted in the tree view.
4. Click **Reverse**.
5. Oracle Data Integrator starts the reverse-engineering process on the selected data target. The reverse-engineered data targets appear under the Model.
6. Perform steps 1 to 5 to reverse-engineer the following objects:
 - a. InfoCube 0SD_C04

- b. InfoObject 0SOLD_TO
- c. InfoObject 0CREDITOR
- d. InfoObject 0GL_ACCOUNT
- e. ODS/DSO 0BBP_PO
- f. Open Hub destination ZOHD_SALE (/BIC/OHZOHD_SALE)

Reverse-Engineering Without GUI

If you want to perform a reverse-engineering without using the SAP BW Metadata Browser, you set the corresponding parameters in the SAP BW KM . For example, for Info Cube, enter the following parameters in the SAP BW KM options:

- USE_GUI: No
- INFOCUBE_NAME : 0FIAP_C20

With this configuration, the Info Cube data target will be reverse-engineered without displaying the SAP BW Metadata Browser.

By using the appropriate KM options, you can also select a filter for the reversed data targets based on for example an Info object, or a ODS/DSO, or an Open Hub table.

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 BW data targets from Oracle Data Integrator.

What happens when you reverse-engineer SAP BW data targets?

First, Oracle Data Integrator uploads ABAP code in the SAP BW system allowing for retrieving the SAP BW data targets metadata. This RFC is uploaded only if the `UPLOAD_ABAP_CODE` and `UPLOAD_ABAP_BASE` options are selected (this option is 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 BW, Oracle Data Integrator is able to retrieve the SAP BW metadata and to display it in the SAP BW Metadata Browser. Selected metadata is then reverse-engineered from SAP BW into the Oracle Data Integrator repository. The repository only contains the description of the SAP BW metadata, and no data at all.

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

Creating and Running the Mapping

This chapter describes how to create and run the mapping.

This chapter includes the following sections:

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

Creating the Mapping

Now that the ODI source and target data models have been created, it is possible to create the ODI mappings to extract data from the following SAP BW data targets:

- The Info Cube 0FIAP_C20 data target to W_VENDR_BAL_DS Oracle table.
- The Info Object 0CREDITOR data target to W_ACCNT_VENDR_DS Oracle table.
- The Hierarchy data from Info Object 0GL_ACCOUNT to W_GL_ACCOUNT_HIER_DS Oracle table.
- The Interval Hierarchy data from InfoObject 0GL_ACCOUNT to TARGET_INTERVAL_HIERARCHY Oracle table.
- The ODS/DSO 0BBP_PO data target to W_PURCH_DS Oracle table.
- The Info Cube 0FIAP_C20 and Info Object 0CREDITOR data targets join to W_VENDER_CUBE_OBJ_DS Oracle table.
- The InfoCube and Navigation Attribute data from IC 0SD_C04 / IO 0SOLD_TO to TARGET_NAVIGATIONAL_ATTRIBUTES Oracle table.
- The Open Hub destination ZOHG_SALE (/BIC/OHZOHG_SALE) to W_OPENHUB_SALES_DS Oracle table.
- Delta Extraction can be done on any above said data targets except Info Object. How to perform delta extraction is described in [Chapter 7, "Creating a Package for Delta Extraction."](#)

To create the mapping, perform the following steps:

- [Info Cube Extraction](#)
- [Info Object Extraction](#)
- [Info Object Hierarchy Extraction](#)
- [Info Object Interval Hierarchy Extraction](#)
- [ODS/DSO Extraction](#)
- [Info Cube & Related Info Object Extraction](#)

- [Info Cube Extraction with Navigational Attributes](#)
- [Open Hub Extraction](#)

Info Cube Extraction

This section describes how to create a mapping in ODI in order to load data from the SAP BW source model representing the InfoCube 0FIAP_C20 data target to the W_VENDR_BAL_DS Oracle target model.

Create the Mapping

To create the mapping:

1. Open Designer.
2. In the Projects tree view, expand the SAP BW Demo Project.
3. Expand the SAP_BW_DataTargets folder.
4. Select the Mappings node.
5. Right-click and select **New Mapping**.
6. In the Mapping Definition tab, enter the mapping name: InfoCube_SAP_BW_VendorBal.W_VENDR_BAL_DS.
7. From the **File** menu, click **Save** to save your mapping.

Define the Source and Target Datastores

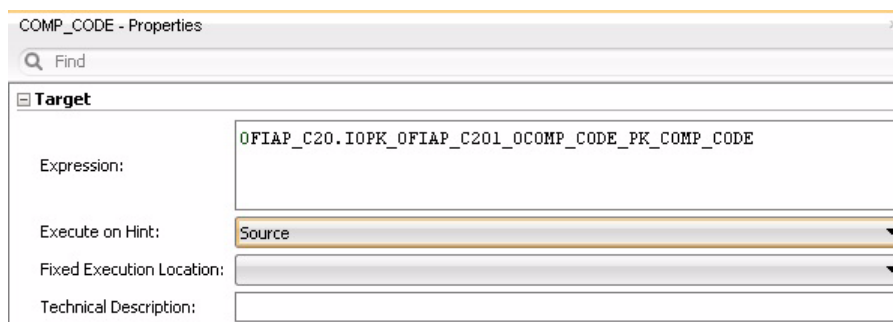
To define the source and target datastores:

1. In the Models tree view, select the W_VENDR_BAL_DS datastore from the Oracle Target Model.
2. Drag this datastore into the Target Datastore area of the Logical diagram (right area of the Logical diagram).
3. In the Models tree view, select the 0FIAP_C20 Vendor Balances datastore from the SAP BW Source model.
4. Drag this datastore into the Sources area of the Logical diagram (left area of the Logical diagram).

Create the Mappings

To create the mappings of the target datastore:

1. Select the 0COMP_CODE attribute from the target datastore.
2. Drag the IOPK_0FIAP_C201_0COMP_CODE_PK_COMP_CODE attribute from the 0FIAP_C20 Vendor Balances source datastore onto the COMP_CODE attribute of the target datastore.
3. Make sure that the **Execute on Hint** is set to **Source**. The mapping should look as shown below:



0COMP_CODE

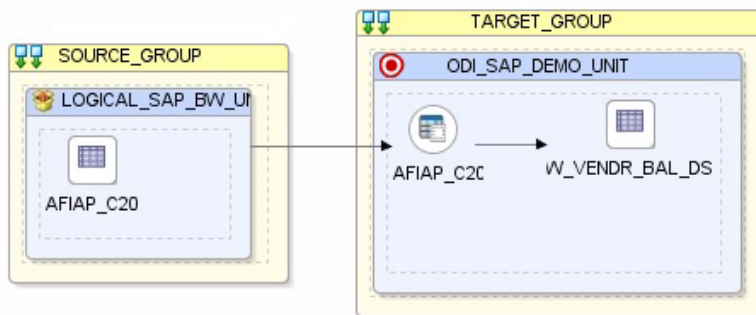
4. Repeat this operation to perform the following simple mappings:

- W_VENDR_BAL_DS.OCREDITOR = 0FIAP_C20.IOPK_0FIAP_C202_0CREDITOR_PK_VENDOR
- W_VENDR_BAL_DS.ORG ACCOUNT = 0FIAP_C20.IOPK_0FIAP_C202_0RC_ACCOUNT_PK_GL_ACCOUNT
- W_VENDR_BAL_DS.TOTAL DEBIT POSTING = 0FIAP_C20.KF_0DEBIT
- W_VENDR_BAL_DS.TOTAL CREDIT POSTING - 0FIAP_C20.KF_0DEBIT
- W_VENDR_BAL_DS.CUMULATIVE BALANCE - 0FIAP_C20.KF_0BALANCE

Define the Mapping Flow

To define the mapping flow:

1. Select the Physical diagram. The Flow diagram looks as shown below:



2. In the target group, select the access point for the source group that contains Info Cube 0FIAP_C20 data target. This source group represents the source dataset made up of SAP BW data target.
3. In the Property panel, under Loading Knowledge Module, select LKM SAP BW to Oracle (SQLDR) or LKM SAP BW to SQL from the Loading Knowledge Module drop-down list.
4. In the target group, select the target Oracle database, which is also used as the staging area for this mapping.
5. In the 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.

Info Object Extraction

This section describes how to create a mapping in ODI in order to load data from a SAP BW source model representing the Info Object 0CREDITOR data target to the W_ACCNT_VENDR_DS Oracle target model.

Create the Mapping

1. Open Designer.
2. In the Projects tree view, expand the SAP BW Demo Project.
3. Expand the SAP_BW_DataTargets folder.
4. Select the Mappings node.
5. Right-click and select **New Mapping**.
6. In the Mapping Definition tab, enter the mapping name: InfoObject_SAP_BW_VendorAcc.W_ACCNT_VENDR_DS.
7. From the **File** menu, click **Save** to save your mapping.

Define the Source and Target Datastores

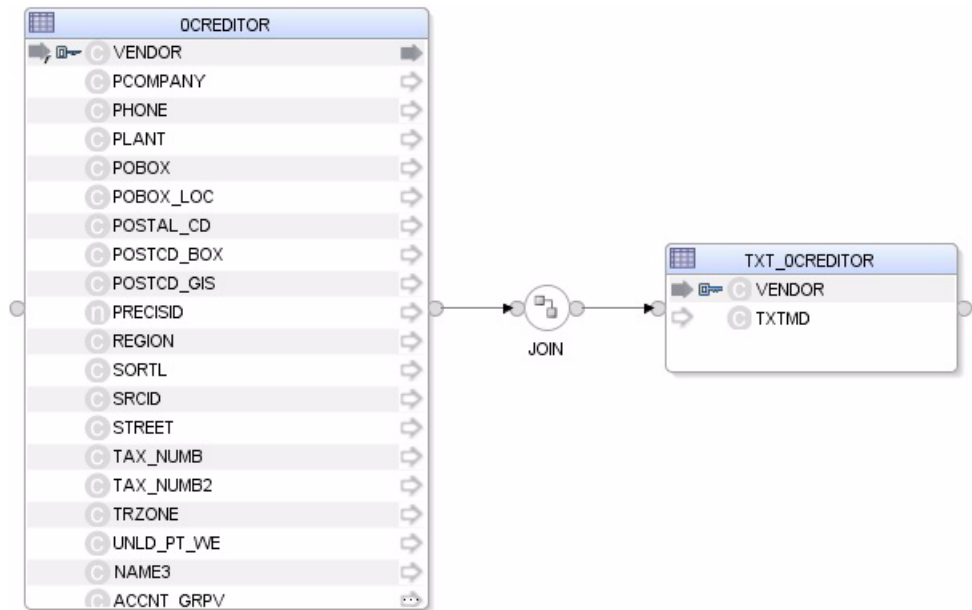
To define the source and target datastores:

1. In the Models tree view, select the W_ACCNT_VENDR_DS datastore from the Oracle Target Model.
2. Drag this datastore into the Target Datastore area of the Logical diagram (right area of the Logical diagram).
3. In the Models tree view, select the Info Object's main table 0CREDITOR Vendor datastore from the SAP BW Source model.
4. Drag this datastore into the Sources area of the Logical diagram (left area of the Logical diagram).
5. Perform the same operation for the Info Object text table TXT_0CREDITOR Texts: Char.Vendor datastore from the SAP BW Source model.

Define Joins between Info Object's Main table and Text Table

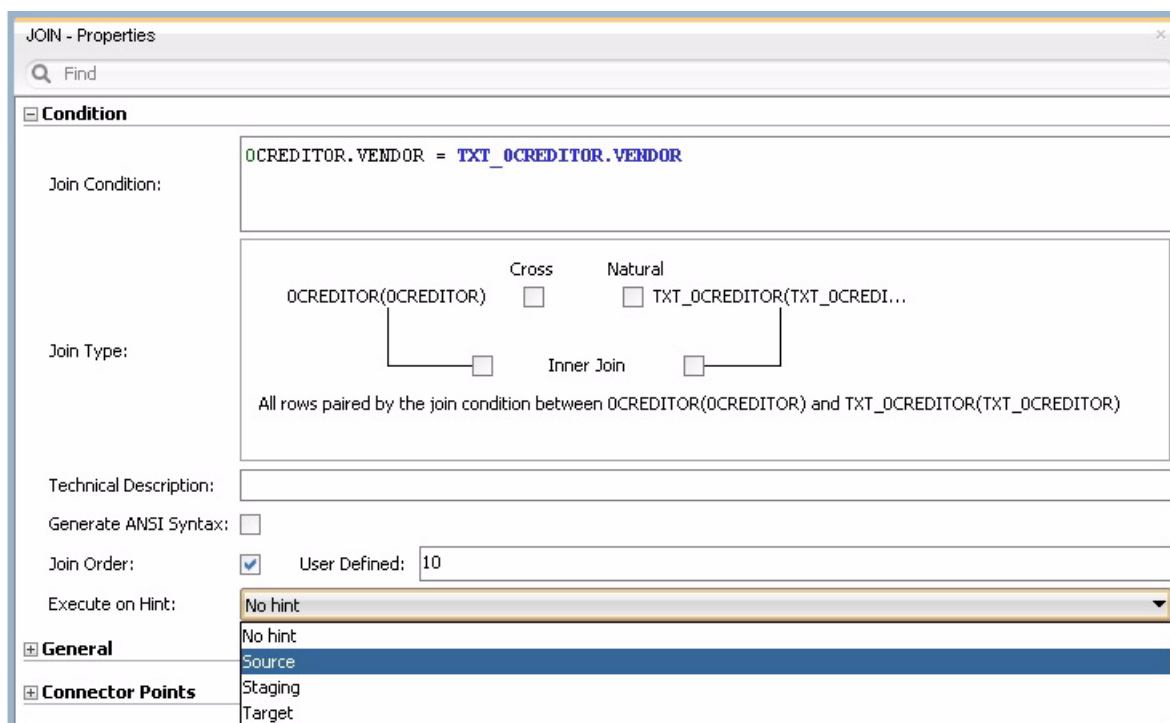
To create joins between the source datastores of Info Object's main table and text table in the mapping:

1. In the Logical diagram when you drag the Info Object's text table TXT_0CREDITOR Texts: Char.Vendor the join will be automatically created on attribute vendor from the 0CREDITOR Vendor Info object's main table datastore and the Vendor attribute of Info Object's text table TXT_0CREDITOR Texts: Char.Vendor. This defines the first join on the vendor Key identifier, as shown below:



0CREDITOR Vendor Info object's main table datastore is shown. On the right the Vendor attribute of Info Object's text table TXT_0CREDITOR Texts: Char. Vendor

2. In the Properties panel, select Join Order and make sure that the **Execute on Hint** option is set to **Source** as shown below:

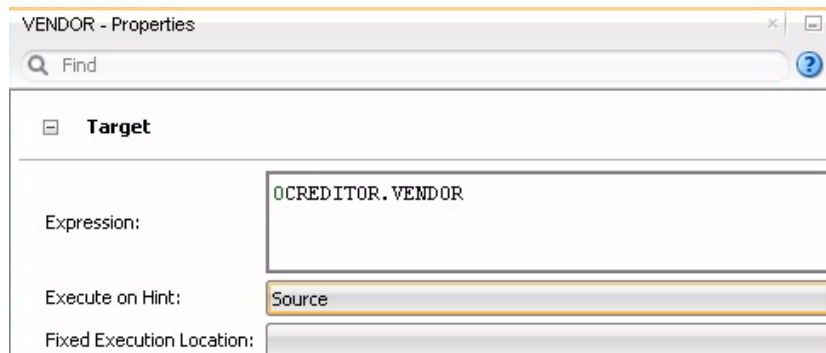


3. Make sure that table number of main table 0CREDITOR is smaller than that of text table TXT_0CREDITOR. In *1:n relationships* the parent table has to be the first in a join.
4. Change join type to left outer join (All rows from 0CREDITOR and matching ones from TXT_0CREDITOR).

Create the Mappings

To create the mappings of the target datastore:

1. Select the `VENDOR` attribute from the target datastore.
2. Drag the `VENDOR` attribute from the `OCREDITOR` source data target onto the `VENDOR` attribute of the target datastore.
3. Make sure that the **Execute on Hint** option is set to **Source**. The mapping should look as shown below:

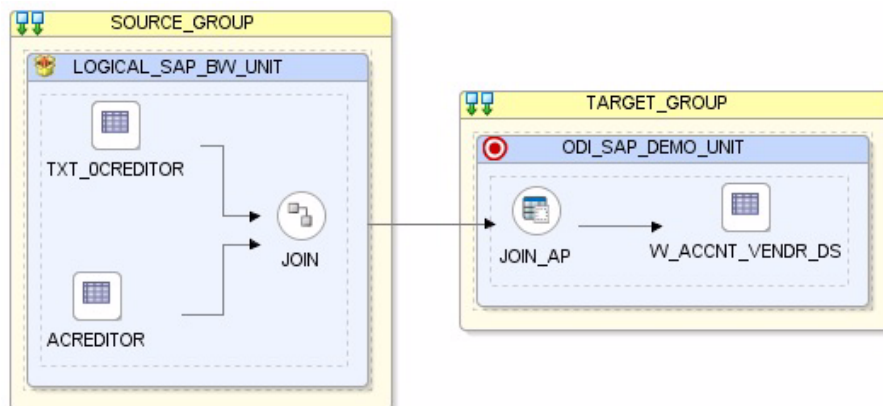


4. Repeat this operation to perform the following simple mappings:
 - `W_ACCNT_VENDR_DS.PCOMPANY = OCREDITOR.PCOMPANY`
 - `W_ACCNT_VENDR_DS.PHONE = OCREDITOR.PHONE`
 - `W_ACCNT_VENDR_DS.PLANT = OCREDITOR.PLANT`
 - `W_ACCNT_VENDR_DS.STREET = OCREDITOR.STREET`
 - `W_ACCNT_VENDR_DS.MED_DESC = TXT_OCREDITOR.TXTMD`

Define the Mapping Flow

To define the mapping flow:

1. Select the Physical diagram. The Flow diagram looks as shown below:



2. In the target group, select the access point for the source group that contains both the Info Object's main table `OCREDITOR` and text table `TXT_OCREDITOR`. This source group represents the source dataset made up of Info Object's main table & text table joined `SAP BW` data targets.

3. In the Property panel, under Loading Knowledge Module, select `LKM SAP BW to Oracle (SQLLDR)` or `LKM SAP BW to SQL` from the Loading Knowledge Module drop-down list.
4. In the target group, select the target Oracle database, which is also used as the staging area for this mapping.
5. In the Property panel, under the 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.

Info Object Hierarchy Extraction

This section describes how to create a mapping in ODI in order to load data from a SAP BW source model representing the hierarchy data for Info Object `0GL_ACCOUNT` data target to the `W_GL_ACCOUNT_HIER_DS` Oracle target model.

Create the Mapping

1. Open Designer.
2. In the Projects tree view, expand the `SAP BW Demo Project`.
3. Expand the `SAP_BW_DataTargets` folder.
4. Select the Mappings node.
5. Right-click and select **New Mapping**.
6. In the Mapping Definition tab, enter the mapping name: `InfoObject_SAP_BW_GLAccount.W_GL_ACCOUNT_HIER_DS`.
7. From the **File** menu, click **Save** to save your mapping.

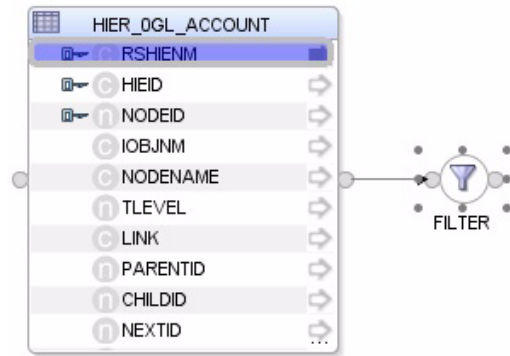
Define the Source and Target Datastores

To define the source and target datastores:

1. In the Models tree view, select the `W_GL_ACCOUNT_DS` datastore from the `Oracle Target Model`.
2. Drag this datastore into the target area of the Logical diagram (right area of the Logical diagram).
3. In the Models tree view, select the Info Object's hierarchy table `HIER_0GL_ACCOUNT` datastore from the `SAP BW Source model`.
4. Drag this datastore into the sources area of the Logical diagram (left area of the Logical diagram).
5. Add the following filter condition on the `RSHIENM` source attribute: `HIER_0GL_ACCOUNT.RSHIENM = 'ACI1'`

Please note that `ACI1` is the name of a hierarchy defined for `0GL_ACCOUNT`. A list of defined hierarchies is available under **Description** section of the **Definition** tab when opening the hierarchy data store.

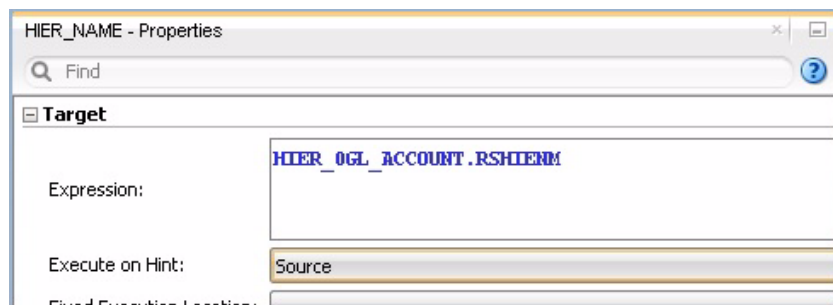
6. After adding the filter condition the Sources diagram should look as shown below:



Create the Mappings

To create the mappings of the target datastore:

1. Select the HIER_NAME attribute from the target datastore.
2. Drag the RSHIENM attribute from the HIER_OGL_ACCOUNT source data target onto the HIER_NAME attribute of the target datastore.
3. Make sure that the **Execute on Hint** option is set to **Source**. The mapping should look as shown below:

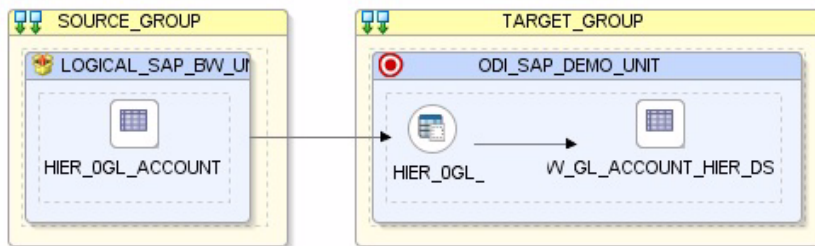


4. Repeat this operation to perform the following simple mappings:
 - W_GL_ACCOUNT_HIER_DS.NODEID = HIER_OGL_ACCOUNT.NODEID
 - W_GL_ACCOUNT_HIER_DS.INFOOBJECT_NAME = HIER_OGL_ACCOUNT.IOBJNM
 - W_GL_ACCOUNT_HIER_DS.NODENAME = HIER_OGL_ACCOUNT.NODENAME
 - W_GL_ACCOUNT_HIER_DS.TLEVEL = HIER_OGL_ACCOUNT.TLEVEL
 - W_GL_ACCOUNT_HIER_DS.PARENTID = HIER_OGL_ACCOUNT.PARENTID
 - W_GL_ACCOUNT_HIER_DS.CHILDDID = HIER_OGL_ACCOUNT.CHILDDID
 - W_GL_ACCOUNT_HIER_DS.NEXTID = HIER_OGL_ACCOUNT.NEXTID

Define the Mapping Flow

To define the mapping flow:

1. Select the Physical diagram. The Flow diagram looks as shown below:



2. In the target group, select the access point for the source group that contains both the Info Object's main table 0CREDITOR and text table TXT_0CREDITOR. This source group represents the source dataset made up of Info Object's main table & text table joined SAP BW data targets.
3. In the Property panel, under Loading Knowledge Module, select LKM SAP BW to Oracle (SQLLDR) or LKM SAP BW to SQL from the Loading Knowledge Module drop-down list.
4. In the target group, select the target Oracle database, which is also used as the staging area for this mapping.
5. In the 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.

Info Object Interval Hierarchy Extraction

This section describes how to extract data from InfoObject and InfoObject hierarchy table containing an interval hierarchy. The data is extracted from InfoObject OGL_ACCOUNT and InfoObject hierarchy table HIER_OGL_ACCOUNT.

Create the Mapping

1. Open Designer.
2. In the Projects tree view, expand the SAP BW Demo Project.
3. Expand the SAP_BW_DataTargets folder.
4. Select the Mappings node.
5. Right-click and select **New Mapping**.
6. In the Mapping Definition tab, enter the mapping name: InfoObject_InfoObjectHierarchy_join.OGL_ACCOUNT.
7. From the **File** menu, click **Save** to save your mapping.

Define the Source and Target Datastores

To define the source and target datastores:

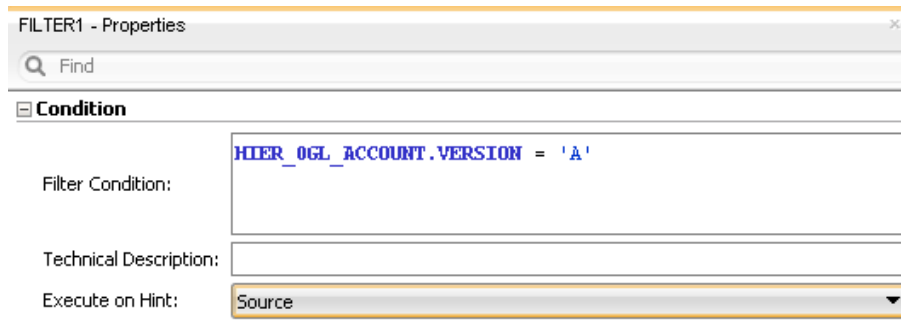
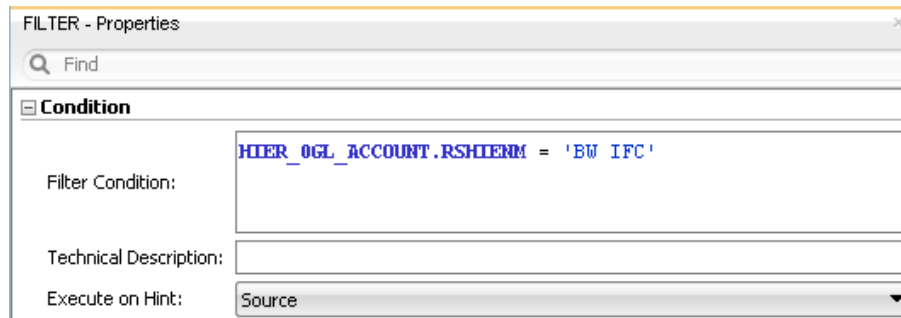
1. In the Models tree view, select the TARGET_INTERVAL_HIERARCHY datastore from the Oracle_Target Model W_SD_DELIVERY_CUBE_SOLDTO.

2. Drag this datastore into the target area of the Logical diagram (right area of the Logical diagram).
3. In the Models tree view, select the InfoObject 0GL_ACCOUNT from the SAP BW source model.
4. Drag this datastore into the sources area of the Logical diagram (left area of the Logical diagram).
5. Perform the same operation for the InfoObject hierarchy table HIER_0GL_ACCOUNT datastore from the SAP BW Source model.

Define Joins between Sources (Info Object Main table and InfoObject Hierarchy Table)

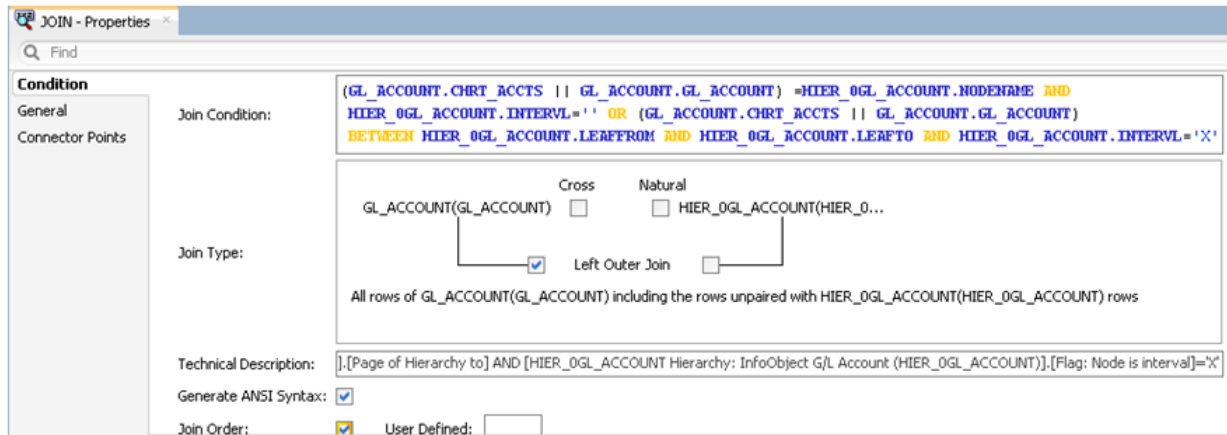
The join between InfoObject and InfoObject Hierarchy table may automatically be added by ODI. - There may be situations when more than one join shows up. E.g. when the same InfoObject is referenced more than once in different characteristics. Such unwanted extra joins have to be removed.

Apply **source-side** Filters on RSHIENM and VERSION:

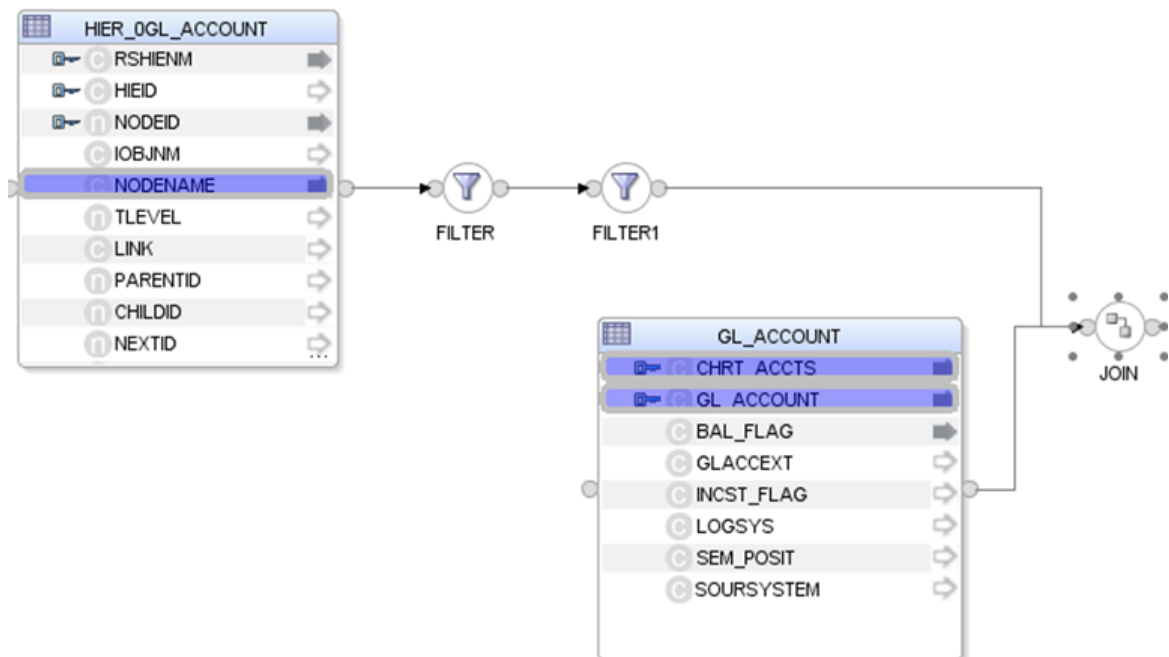


Add joins: all joins must be executed on *staging* area. Not on source.

The join expression looks like this:



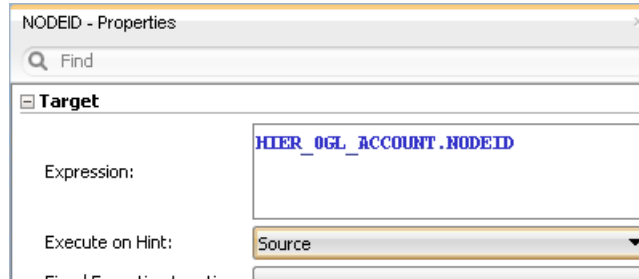
The final join and filter setup should look like this:



Create the Mappings

To create the mappings of the target datastore:

1. Drag the **NODEID** attribute from the **HIER_OGL_ACCOUNT** source data target onto the **NODEID** attribute of the **TARGET_INTERVAL_HIERARCHY** target datastore.
2. Make sure that the **Execute on Hint** option is set to Source. The mapping should look as shown below:

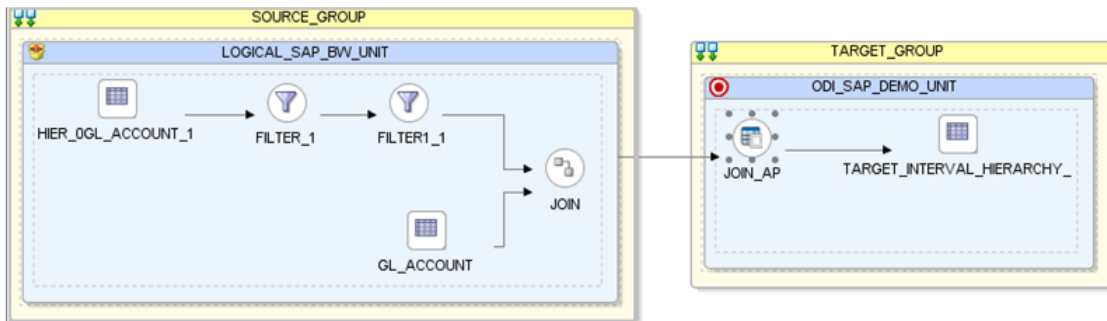


3. Repeat this operation to perform the following simple mappings:
 - TARGET_INTERVAL_HIERARCHY.RSHIENM = HIER_OGL_ACCOUNT.RSHIENM
 - TARGET_INTERVAL_HIERARCHY.PARENTID = HIER_OGL_ACCOUNT.PARENTID
 - TARGET_INTERVAL_HIERARCHY.INTERVL = HIER_OGL_ACCOUNT.INTERVAL
 - TARGET_INTERVAL_HIERARCHY.LEAFFROM = HIER_OGL_ACCOUNT.LEAFFROM
 - TARGET_INTERVAL_HIERARCHY.LEAFTO = HIER_OGL_ACCOUNT.LEAFTO
 - TARGET_INTERVAL_HIERARCHY.VERSION = HIER_OGL_ACCOUNT.VERSION
 - TARGET_INTERVAL_HIERARCHY.CHRT_ACCTS = OGL_ACCOUNT.CHRT_ACCTS
 - TARGET_INTERVAL_HIERARCHY.GL_ACCOUNT = OGL_ACCOUNT.GL_ACCOUNT
 - TARGET_INTERVAL_HIERARCHY.BAL_FLAG = OGL_ACCOUNT.BAL_FLAG
 - TARGET_INTERVAL_HIERARCHY.GLACCEXT = OGL_ACCOUNT.GLACCEXT
 - TARGET_INTERVAL_HIERARCHY.INCST_FLAG = OGL_ACCOUNT.INCST_FLAG

Define the Mapping Flow

To define the mapping flow:

1. Select the Physical diagram. The Flow diagram looks as shown below:



2. In the target group, select the access point for the ODI Source Group.
3. In the Property panel, under Loading Knowledge Module, verify that LKM SAP BW to Oracle (SQLLDR) or LKM SAP BW to SQL has been set.
4. In the target group, select the target Oracle datastore, which is also used as the staging area for this mapping.
5. In the Property panel, under Integration Knowledge Module, verify that IKM SQL Control Append has been set.
6. Execute the Mapping. Verify that data got populated into the target datastore.

ODS/DSO Extraction

This section describes how to create a mapping in ODI in order to load data from a SAP BW source model representing the ODS/DSO 0BBP_PO data target to the W_PURCH_DS Oracle target model.

Create the Mapping

1. Open Designer.
2. In the Projects tree view, expand the SAP BW Demo Project.
3. Expand the SAP_BW_DataTargets folder.
4. Select the Mappings node.
5. Right-click and select **New Mapping**.
6. In the Mapping Definition tab, enter the mapping name: ODS_SAP_BW_PurchOrd1.W_PURCH_DS.
7. From the **File** menu, click **Save** to save your mapping.

Define the Source and Target Datastores

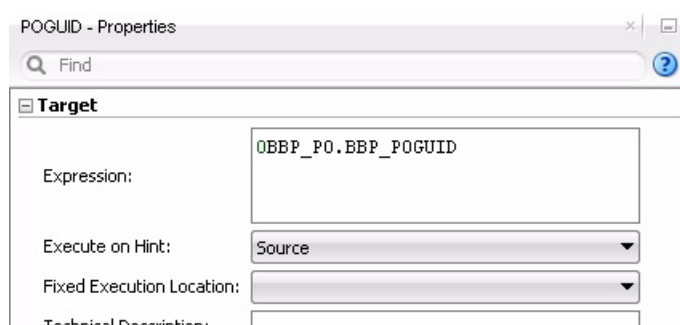
To define the source and target datastores:

1. In the Models tree view, select the W_PURCH_DS datastore from the Oracle_Target Model.
2. Drag this datastore into the targets area of the Logical diagram (right area of the Logical diagram).
3. In the Models tree view, select the 0BBP_PO Purchase Order - Single Documents datastore from the SAP BW Source model.
4. Drag this datastore into the sources area of the Logical diagram (left area of the Logical diagram).

Create the Mappings

To create the mappings of the target datastore:

1. Select the PO_GUIDE attribute from the target datastore.
2. Drag the BBP_POGUID attribute from the 0BBP_PO Purchase Order - Single Documents source data target onto the PO_GUIDE attribute of the target datastore.
3. Make sure that the **Execute on Hint** option is set to **Source**. The mapping should look as shown below:



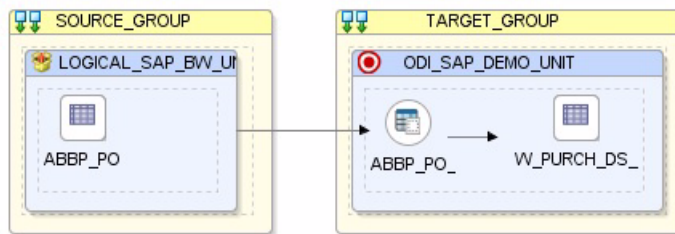
4. Repeat this operation to perform the following simple mappings:

- W_PURCH_DS.ORDER_ITEM_GUID = 0BBP_PO.BBP_POITGU
- W_PURCH_DS.ACC_ASS_GUID = 0BBP_PO.BBP_ACGUID
- W_PURCH_DS.GUIDE_CONTRCT = 0BBP_PO.BBP_CTGUID
- W_PURCH_DS.ORDER_QUANT = 0BBP_PO.BBP_ASPQOU
- W_PURCH_DS.NET_PRICE = 0BBP_PO.BBP_NETPRC
- W_PURCH_DS.PRICE_UNIT = 0BBP_PO.BBP_PRCUNT
- W_PURCH_DS.NO_OF_PURCH_ODR = 0BBP_PO.BBP_POCT

Define the Mapping Flow

To define the mapping flow:

1. Select the Physical diagram. The Flow diagram looks as shown below:



2. In the target group, select the access point for the source group that contains ODS/DSO 0BBP_PO data targets. This source group represents the source dataset made up of SAP BW data target.
3. In the Property panel, under Loading Knowledge Module, select LKM SAP BW to Oracle (SQLLDR) or LKM SAP BW to SQL from the Loading Knowledge Module drop-down list.
4. In the target group, select the target Oracle database, which is also used as the staging area for this mapping.
5. In the 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.

Info Cube & Related Info Object Extraction

This section describes how to create a mapping in ODI in order to load data from a SAP BW source model representing the Info Cube 0FIAP_C20 and Info Object 0CREDITOR data targets join to the W_VENDER_CUBE_OBJ_DS Oracle target model.

Create the Mapping

1. Open Designer.
2. In the Projects tree view, expand the SAP BW Demo Project.
3. Expand the SAP_BW_DataTargets folder.

4. Select the Mappings node.
5. Right-click and select **New Mapping**.
6. In the Mapping Definition tab, enter the mapping name: InfoCube_InfoObject_join.W_VENDER_CUBE_OBJ_DS.
7. From the **File** menu, click **Save** to save your mapping.

Define the Source and Target Datasources

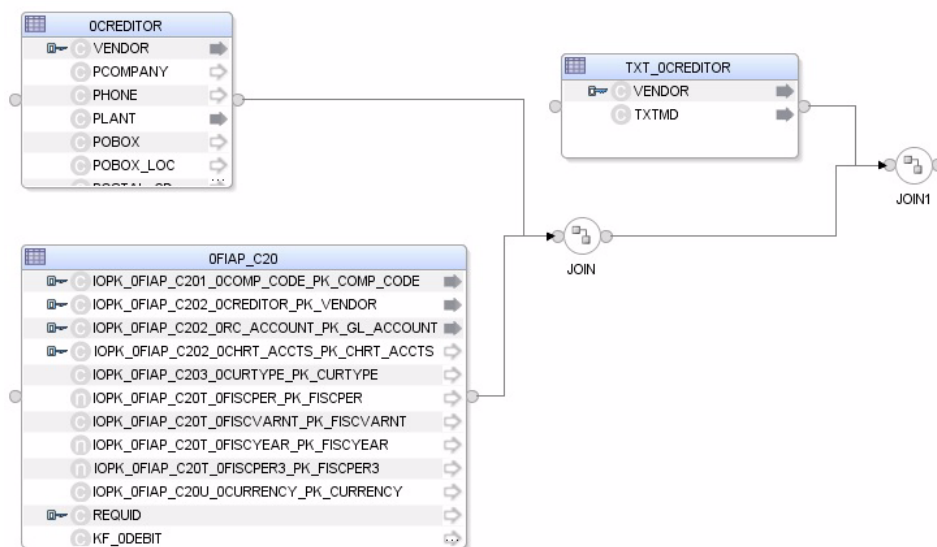
To define the source and target datasources:

1. In the Models tree view, select the W_VENDER_CUBE_OBJ_DS datasource from the Oracle_Target Model.
2. Drag this datasource into the target area of the Logical diagram (right area of the Logical diagram).
3. In the Models tree view, select the Info Cube 0FIAP_C20 Vendor Balances datasource from the SAP BW Source model.
4. Drag this datasource into the sources area of the Logical diagram (left area of the Logical diagram).
5. Perform the same operation for the Info Object's main table 0CREDITOR Vendor and text table TXT_0CREDITOR Texts: Char. Vendor datasource from the SAP BW Source model.

Define Joins between Sources (Info Cube & Object)

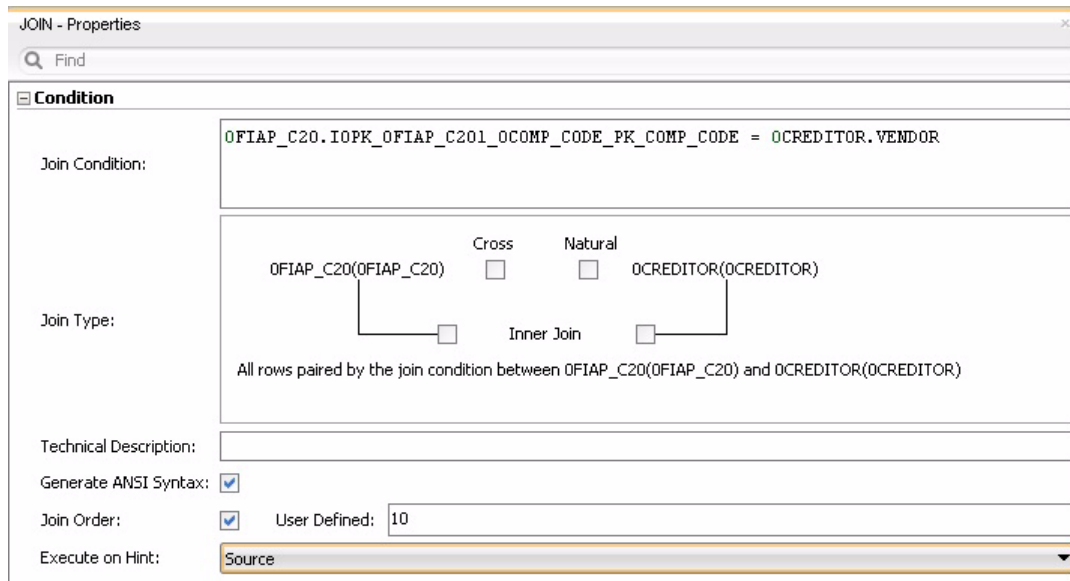
To create joins between the source datasources of a mapping:

1. In the Sources area of the Logical diagram drag the IOPK_0FIAP_C202_0CREDITOR_PK_VENDOR attribute from the Info Cube 0FIAP_C20 Vendor Balances datasource onto the VENDOR attribute in the Info Object's main table 0CREDITOR Vendor. This defines the first join on the Vendor Key identifier, as shown below:



2. In the Properties panel, you can see the join clause: 0FIAP_C20.IOPK_0FIAP_C202_0CREDITOR_PK_VENDOR=0CREDITOR.VENDOR

3. In the Properties panel, select **Generate ANSI Syntax** and **Join Order** options, and make sure that **Execute on Hint** option is set to **Source** as shown below:

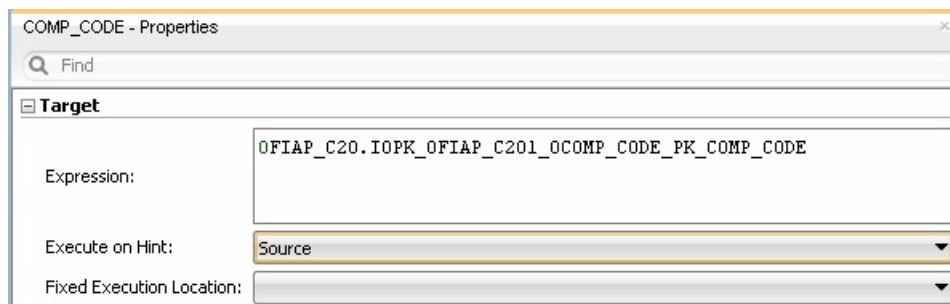


4. Make sure that table number of 0FIAP_C20 is smaller than that of 0CREDITOR. In *1:n relationships* the parent data target has to be the first in a join.
5. Change join between 0FIAP_C20 and 0CREDITOR to left outer join (All rows from 0FIAP_C20 and matching ones from 0CREDITOR).
6. Change join between 0CREDITOR and TXT_0CREDITOR to left outer join (All rows from 0CREDITOR and matching ones from TXT_0CREDITOR).

Create the Mappings

To create the mappings of the target datastore:

1. Select the COMP_CODE attribute from the target datastore.
2. Drag the IOPK_0FIAP_C201_0COMP_CODE_PK_COMP_CODE attribute from the 0FIAP_C20 Vendor Balances source data target onto the COMP_CODE attribute of the target datastore.
3. Make sure that the **Execute on Hint** option is set to **Source**. The mapping should look as shown below:



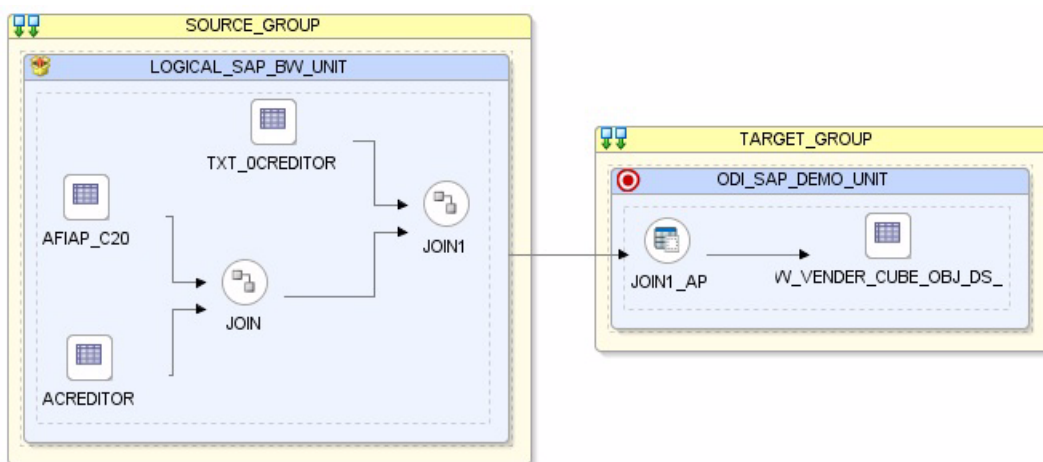
4. Repeat this operation to perform the following simple mappings:
 - W_VENDER_CUBE_OBJ_DS.VENDOR = 0FIAP_C20.IOPK_0FIAP_C202_0CREDITOR_PK_VENDOR

- W_VENDER_CUBE_OBJ_DS.GL_ACCOUNT = 0FIAP_C20.IOPK_0FIAP_C202_0RC_ACCOUNT_PK_GL_ACCOUNT
- W_VENDER_CUBE_OBJ_DS.SALES_FOR_PERIOD = 0FIAP_C20.KF_0SALES
- W_VENDER_CUBE_OBJ_DS.CUM_BALANCE = 0FIAP_C20.KF_0BALANCE
- W_VENDER_CUBE_OBJ_DS.PLANT = 0CREDITOR.PLANT
- W_VENDER_CUBE_OBJ_DS.MEDIUM_DESC = TXT_0CREDITOR.TXTMD

Define the Mapping Flow

To define the mapping flow:

1. Select the Physical diagram. The Flow diagram looks as shown below:



2. In the target group, select the access point for the source group that contains both the Info Cube and Info Object tables. This source group represents the source dataset made up of two joined SAP BW data targets.
3. In the Property panel, under Loading Knowledge Module, select LKM SAP BW to Oracle (SQLLDR) or LKM SAP BW to SQL from the Loading Knowledge Module drop-down list.
4. In the Property panel, select the target Oracle database, which is also used as the staging area for this mapping.
5. In the 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.

Info Cube Extraction with Navigational Attributes

This section describes how to create a mapping in ODI in order to load data from a SAP InfoCube and its navigation attributes. The data is extracted from Info Cube 0SD_C04 and InfoObject 0SOLD_TO.

Create the Mapping

1. Open Designer.
2. In the Projects tree view, expand the SAP BW Demo Project.
3. Expand the SAP_BW_DataTargets folder.
4. Select the Mappings node.
5. Right-click and select **New Mapping**.
6. In the Mapping Definition tab, enter the mapping name: InfoCube_InfoObject_Nav_join.0SD_C04.
7. From the **File** menu, click **Save** to save your mapping.

Define the Source and Target Datastores

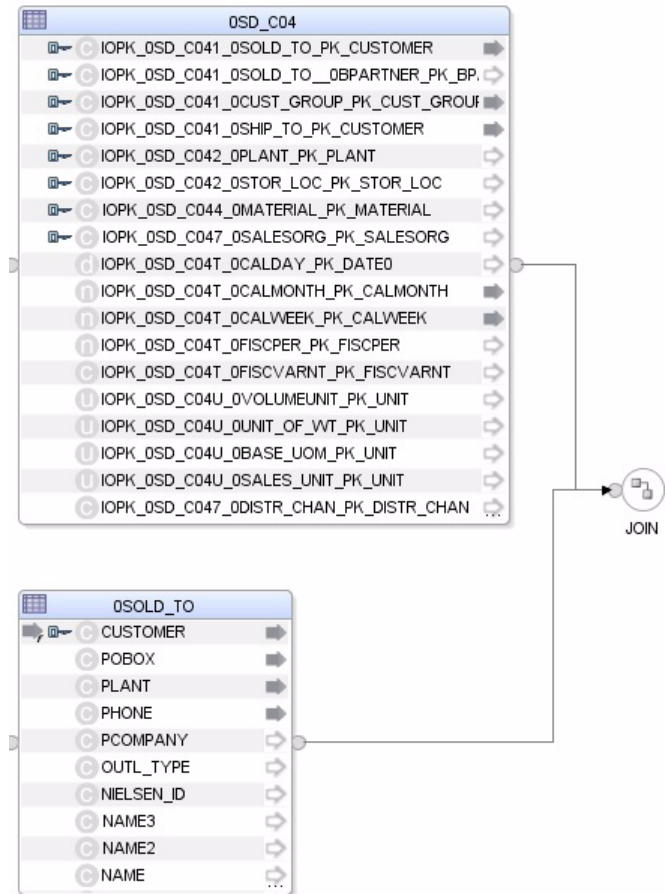
To define the source and target datastores:

1. In the Models tree view, select the 0SD_C04 datastore from the Oracle Target Model TARGET_NAVIGATIONAL_ATTRIBUTES.
2. Drag this datastore into the targets area of the Logical diagram (right area of the Logical diagram).
3. In the Models tree view, select the Info Cube 0SD_C04 Delivery service datastore from the SAP BW Source model.
4. Drag this datastore into the sources area of the Logical diagram (left area of the Logical diagram).
5. Perform the same operation for the Info Object main table 0SOLD_TO Customer datastore from the SAP BW Source model.

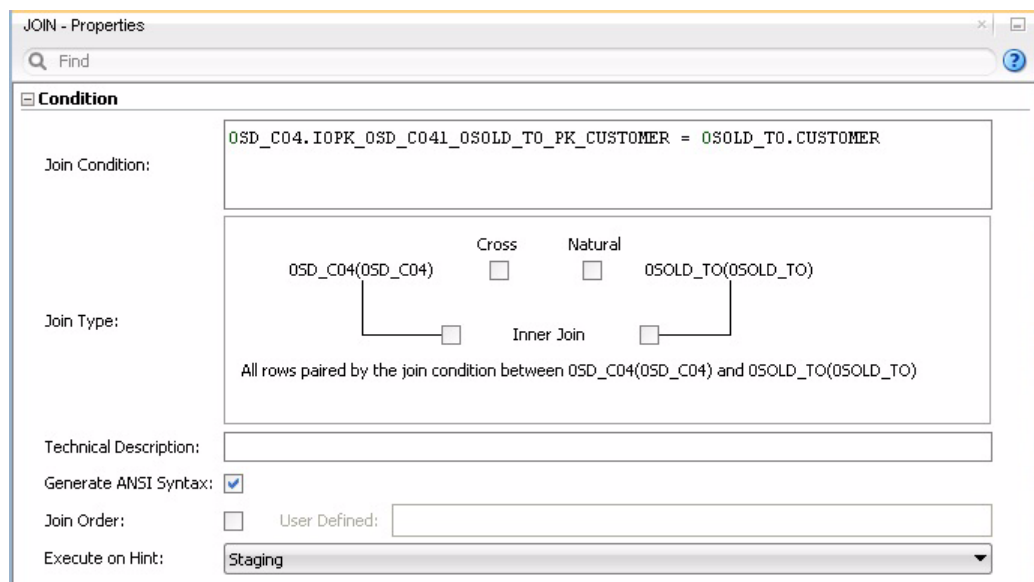
Define Joins Between Sources (Info Cube & Info Object)

The join between InfoCube and InfoObject will automatically be added by ODI. - There may be situations when more than one join shows up. E.g. when the same InfoObject is referenced more than once in different characteristics. Such unwanted extra joins have to be removed.

The final join setup with its join expression should look like this:



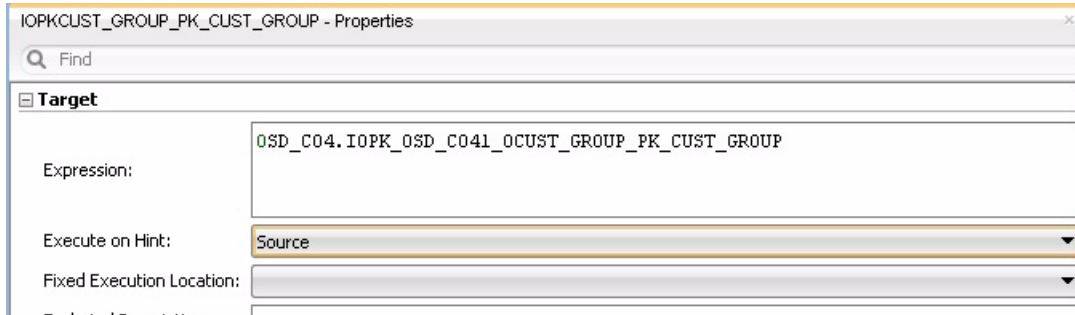
All InfoObject joins for retrieving navigation attributes must be executed on *staging* area. Not on source:



Create the Mappings

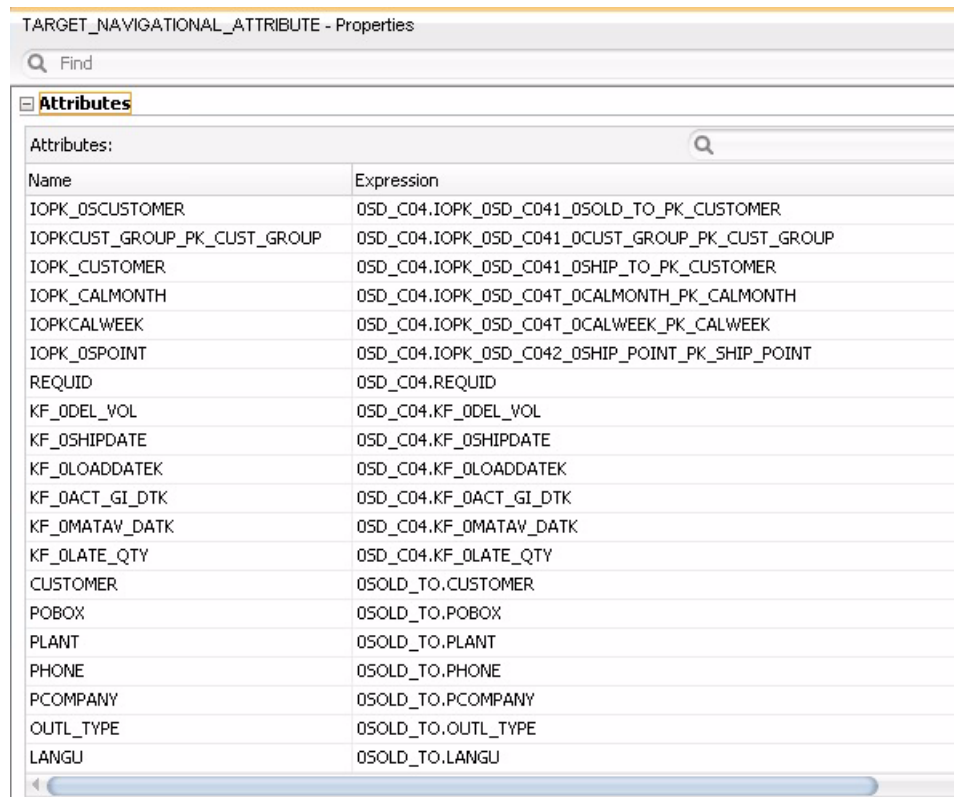
To create the mappings of the target datastore:

1. Drag the `OSD_C04.IOPK_OSD_C041_0SOLD_TO_PK_CUSTOMER` attribute from the `OSD_C041 Delivery service` Source data target onto the `IOPK_OSD_C041_0CUST_GROUP_PK_CUST_GROUP` attribute of the target datastore.
2. Make sure that the **Execute on Hint** option is set to `Source`. The mapping should look as shown below:



`OSD_C04.IOPK_OSD_C041_0SOLD_TO_PK_CUSTOMER`

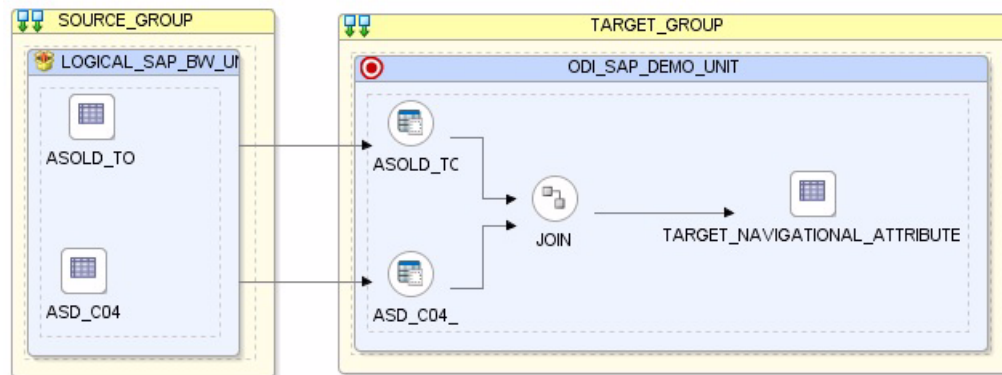
3. Repeat this operation to perform the following simple mappings:



Define the Mapping Flow

To define the mapping flow:

1. Select the Physical diagram. The Flow diagram appears as follows:



2. In the target group, select the access points for the ODI Source Datastores.
3. In the Property panel, verify that LKM SAP BW to Oracle (SQLLDR) or LKM SAP BW to SQL has been set for every ODI Source Datastore.

Note: This mapping will generate several SAP ABAP extraction programs. One for retrieving any display attributes and key figures from the InfoCube and one for each InfoObject containing navigation attributes.

4. In the target group, select the target Oracle database, which is also used as the staging area for this mapping.
5. In the Property panel, under Integration Knowledge Module, verify that IKM SQL Control Append has been set for the target datastore.
6. Execute the Mapping. Verify that Info Cube and Navigational attributes got populated into the target datastore.

Open Hub Extraction

This section describes how to create a mapping in ODI in order to load data from a SAP BW source model representing the Open Hub destination ZOHHD_SALE (/BIC/OHZOHHD_SALE) to the W_OPENHUB_SALES_DS Oracle target model.

Create the Mapping

1. Open Designer.
2. In the Projects tree view, expand the SAP BW Demo Project.
3. Expand the SAP_BW_DataTargets folder.
4. Select the Mappings node.
5. Right-click and select **New Mapping**.
6. In the Mapping Definition tab, enter the mapping name: OpenHub_Sales.W_OPENHUB_SALES_DS.
7. From the **File** menu, click **Save** to save your mapping.

Define the Source and Target Datastores

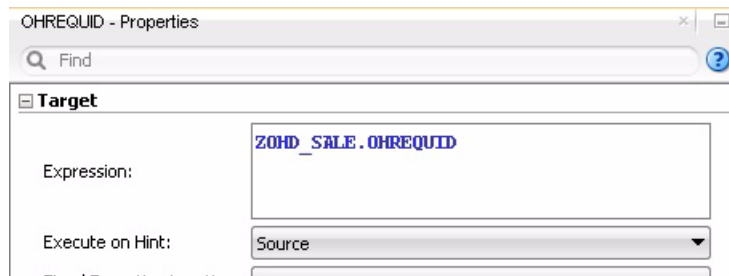
To define the source and target datastores:

1. In the Models tree view, select the W_OPENHUB_SALES_DS datastore from the Oracle_Target Model.
2. Drag this datastore into the targets area of the Logical diagram (right area of the Logical diagram).
3. In the Models tree view, select the ZOHD_SALE datastore from the SAP_BW_Source model.
4. Drag this datastore into the sources area of the Logical diagram (left area of the Logical diagram).

Create the Mappings

To create the mappings of the target datastore:

1. Select the OHREQUID attribute from the target datastore.
2. Drag the OHREQUID attribute from the ZOHD_SALE source data target onto the OHREQUID attribute of the target datastore.
3. Make sure that the **Execute on Hint** option is set to **Source**. The mapping should look as shown below:



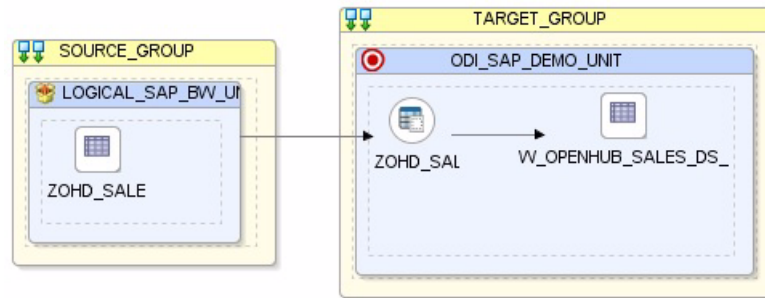
he OHREQUID attribute. It shows that Execute on Hint is set to Source.

4. Repeat this operation to perform the following simple mappings:
 - W_OPENHUB_SALES_DS.DATA_PACKAGE = ZOHD_SALE.DATAPAKID
 - W_OPENHUB_SALES_DS.DATA_RECORD = ZOHD_SALE.RECORD
 - W_OPENHUB_SALES_DS.CUSTOMER = ZOHD_SALE.D_CUSTOMER
 - W_OPENHUB_SALES_DS.MATERIAL = ZOHD_SALE.D_MATERIAL
 - W_OPENHUB_SALES_DS.MATERIAL_PRICE = ZOHD_SALE.PRICE_MAT
 - W_OPENHUB_SALES_DS.CURRENCY = ZOHD_SALE.CURRENCY
 - W_OPENHUB_SALES_DS.QUANTITY = ZOHD_SALE./BIC/ZQTY
 - W_OPENHUB_SALES_DS.TOTAL_REVENUE = ZOHD_SALE./BIC/ZREV

Define the Mapping Flow

To define the mapping flow:

1. Select the Physical diagram. The Flow diagram looks as shown below:



2. In the target group, select the access point for the source group that contains Open Hub Destination Table ZOHD_SALE. This source group represents the source dataset made up of SAP BW Open Hub table.
3. In the Property panel, under Loading Knowledge Module, select LKM SAP BW to Oracle (SQLLDR) or LKM SAP BW to SQL from the Loading Knowledge Module drop-down list.
4. In the target group, select the target Oracle database, which is also used as the staging area for this mapping.
5. In the Property panel, under Integration Knowledge Module, select IKM Oracle Incremental Update from the Integration Knowledge Module drop-down list.
6. Set the IKM options as follows:
 - a. Select No for the *FLOW_CONTROL* option.
 - b. Leave other options as they are.
7. From the **File** menu, click **Save** and close your mapping.

Running the Mapping

This section contains the following topics:

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

Running the Mapping

To run the mapping:

1. In the Projects tree view, expand the SAP Demo Project
2. Expand the SAP_BW_DataTargets folder.
3. Expand the Mappings node.
4. Select the InfoCube_SAP_BW_VendorBal.W_VENDR_BAL_DS mapping.
5. Right-click and select **Run**.
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:

1. It performs the extraction of the data on the SAP BW system. In this example, the extraction will be done on Info Cube source data targets and then returns the 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 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 `InfoCube_SAP_BW_VendorBal.W_VENDR_BAL_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_VENDR_BAL_DS` datastore, right-click and select **Data** to view the data integrated into the target Oracle table.
3. Repeat the same steps for remaining mappings:
 - `InfoObject_SAP_BW_VendorAcc.W_ACCNT_VENDR_DS`
 - `ODS_SAP_BW_PurchOrd1.W_PURCH_DS`
 - `InfoCube_InfoObject_join.W_VENDER_CUBE_OBJ_DS`
 - `OpenHub_Sales.W_OPENHUB_SALES_DS`

Creating a Package for Delta Extraction

This chapter describes how to create a package for delta extraction.

So far we have seen examples of extracting data from BW data targets. The instructions in this chapter explain how the mappings introduced in the previous chapters can be turned into *delta extraction mappings*.

In delta extraction mappings only the data, which was changed after last extraction, is extracted. Delta extraction mappings are only supported for InfoCubes, ODS/DSO and Open Hub objects, but not for InfoObjects.

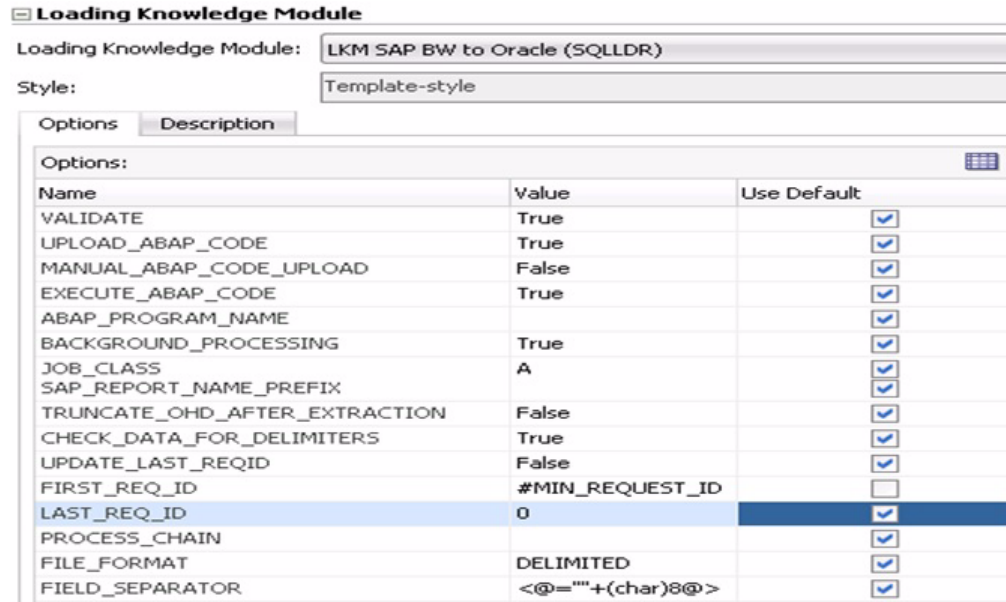
This chapter includes the following sections:

- [Modify the Mapping Flow](#)
- [Creating the MIN_REQUEST_ID Variable](#)
- [Creating a Package for Delta Extraction](#)
- [Running the Package](#)

Modify the Mapping Flow

The first step is to modify the mapping:

1. Select the desired Mappings.
2. Double click the mapping.
3. Select the Physical diagram.
4. In the target group, select the access point for the source group.
5. In the Property panel, under Loading Knowledge Module, set KM option values for FIRST_REQ_ID and LAST_REQ_ID KM as shown below.



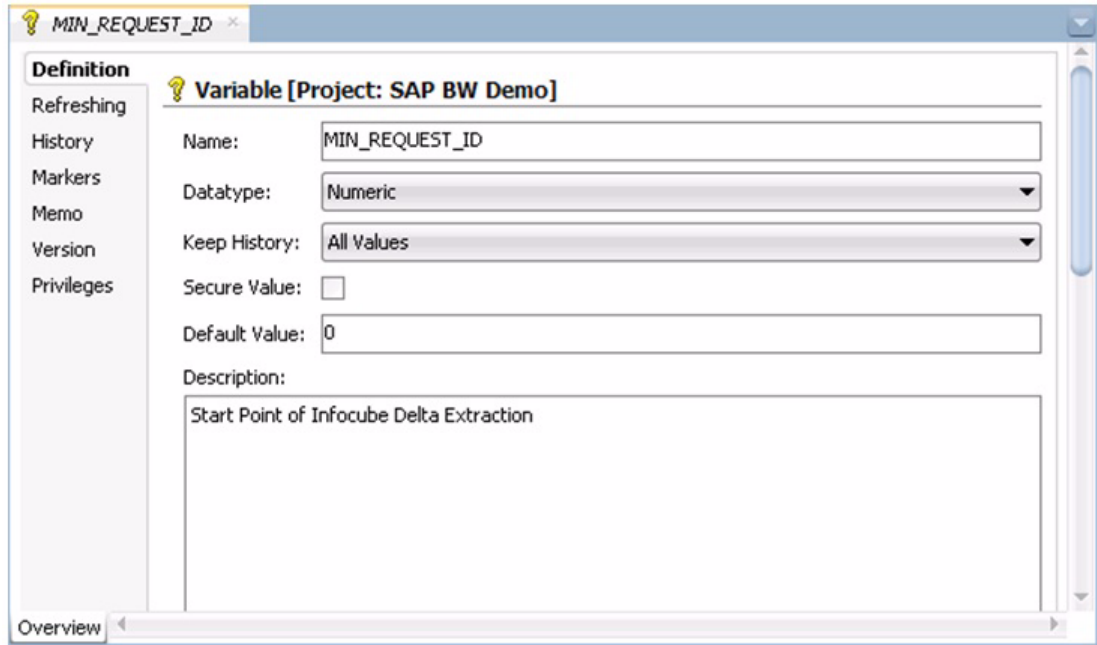
LAST_REQ_ID is highlighted, with a Value of 0 and the Use Default check box selected. The KM option value for FIRST_REQ_ID is #MIN_REQUEST_ID (and the Use Default check box is not selected).

6. From the **File** menu, click **Save** and close your mapping.

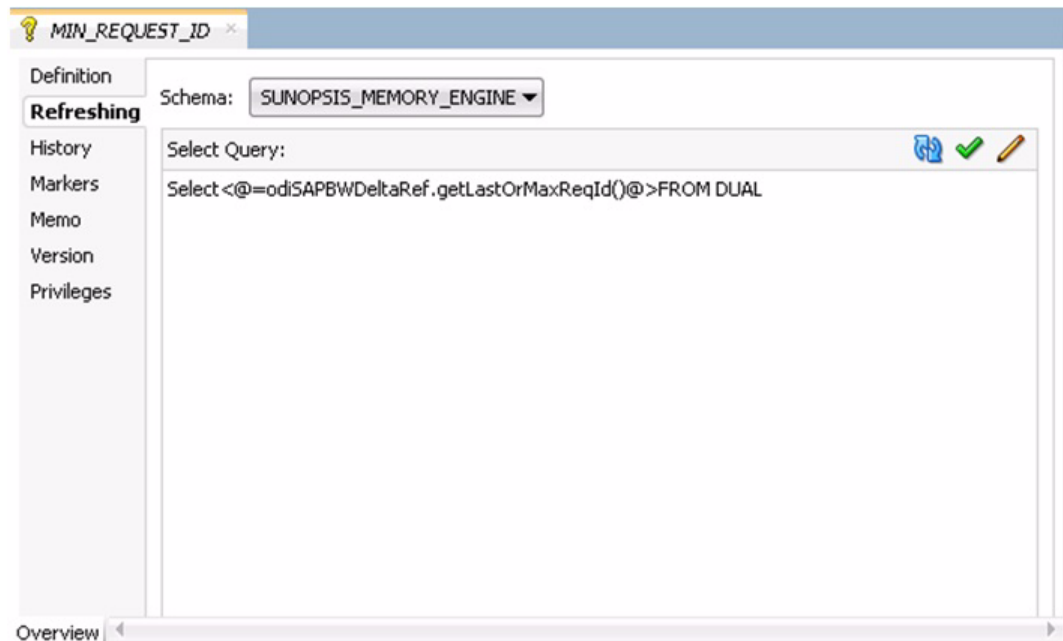
Creating the MIN_REQUEST_ID Variable

The second step consists of creating an ODI variable that stores the last extracted request ID. This request ID serves as the starting point for the next extraction run:

1. Create a MIN_REQUEST_ID variable to hold the last request id. The MIN_REQUEST_ID variable has to be a numeric variable with a Keep History or Last-value Action as shown below.



- The value of this variable is updated at the end of the delta extraction package described in the third step. It happens once delta extraction is finished and the MAX REQUEST ID will be fetched from SAP BW system and will be stored into the ODI Variable (MIN_REQUEST_ID). The refresh expression shown here is used to update the variable:



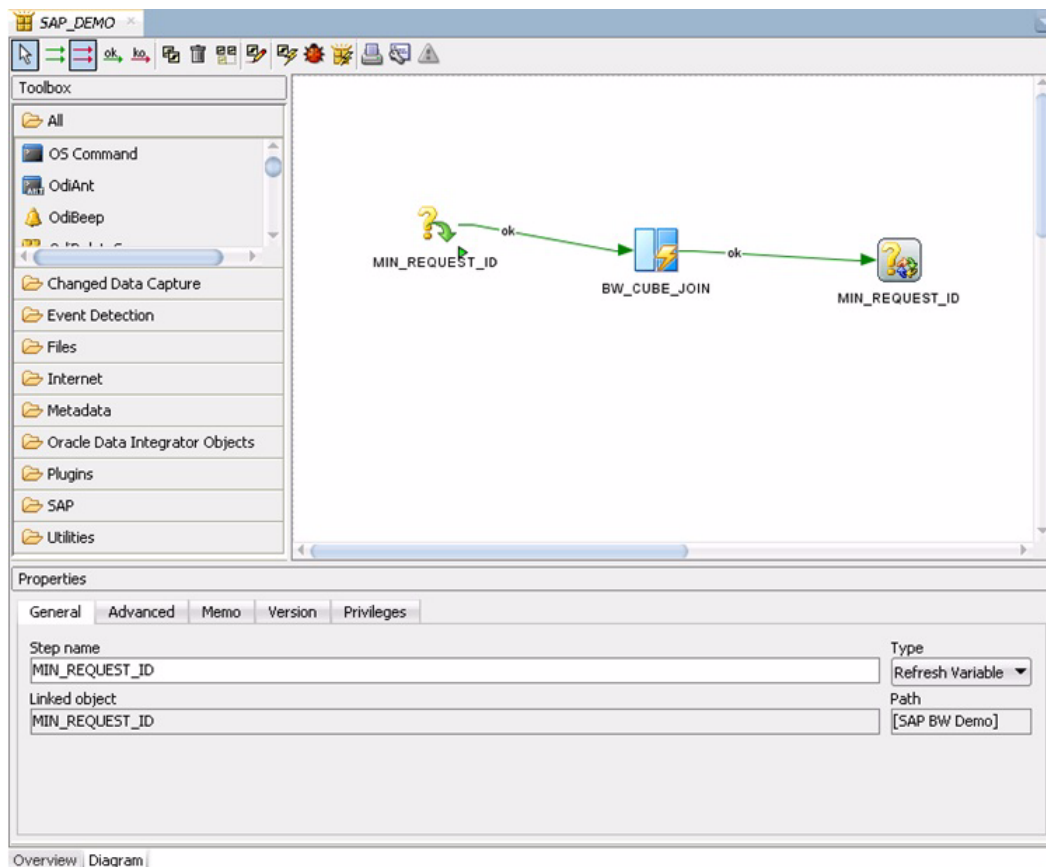
Creating a Package for Delta Extraction

In the third step the building blocks created in the previous steps are put together: Variable and Mappings are arranged into a Package:

- Open the SAP Demo Project.

2. Create a new Package named SAP Demo.
3. Insert a Declare Variable step to this Package:
Drag and Drop the MIN_REQUEST_ID variable and select the "Declare Variable" step type.
4. Insert a Flow step to this Package:
Drag and drop the Mapping created in the previous sections for InfoCube/ ODS or OpenHub-Extraction in the Package.
5. Insert a Refresh Variable step to this Package.
Drag and Drop the MIN_REQUEST_ID variable and select the "Refresh Variable" step type.
6. Define the step sequence.
 - a. Define the Refresh Variable step as the **First Step**.
 - b. Link it to the Flow step that is the next step upon success.
 - c. Link the Flow step to the Refresh Variable step that is the next step upon success and the last step in the Package. Add arrows between the variables.

Your Package should look as shown on this screenshot.



Running the Package

To run the Package:

- In the Projects tree view, expand the SAP Demo Project.

- Select the SAP Demo Package.
- Right-click and select **Run**.
- In the Execution window, click **OK**.
- In the Session Started window, click **OK**.

What happens when you execute the delta extraction Package?

First, Oracle Data Integrator initializes the variable MIN_REQUEST_ID with the last historized value (declare variable step). This value is then passed into ABAP extraction program to specify where to start the data extraction. The KM queries the SAP system for the highest assigned request ID. All data between min and max request ID will be extracted (mapping).

After the execution of the mapping the refresh variable step stores the highest request ID into MIN_REQUEST_ID for next extraction run.

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 BW system.

In this Project, you have:

- Set up the environment and topology to work with SAP BW.
- Created and reverse-engineered a SAP BW data model.
- Created different mappings to load the data from Info Cube, Info Object, ODS/DSO, Info Cube & Info Object join, Open Hub destination table, Delta SAP BW data targets & tables into the Oracle database.

Refer to the *Knowledge Module Developer's Guide for Oracle Data Integrator* for more information on the SAP BW KM features and options.

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 section 3.7.6, "Log Files," in *Application Adapters Guide for Oracle Data Integrator* 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.

SAP ABAP BW Required Privileges

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

For connecting to the SAP system a SAP dialog user is required for RKM installation and LKM execution during development. A SAP RFC user is *not* sufficient.

This user has to 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.

Later for RKM execution and LKM execution in production a 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
Testing ODI Generated Extraction RFCs	SAP Dialog user or SAP RFC user

The following tables list the privileges required for using SAP BW Knowledge Modules:

- [Authorizations Required for RKM SAP BW Upload](#)
- [Authorizations Required for RKM SAP BW Execution](#)
- [Authorizations Required for LKM SAP BW Upload \(Background mode\)](#)
- [Authorizations Required for LKM SAP BW Execution \(Background mode\)](#)
- [Authorizations Required for LKM SAP BW Upload \(Dialog mode\)](#)
- [Authorizations Required for LKM SAP BW Execution \(Dialog mode\)](#)

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 BW Upload

The following authorizations are required for first time installation of **RKM SAP BW**.

Note: Developer Key is required in this authorization.

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
S_RFC	ACTVT	16	
	RFC_NAME	RFC1, RS*, S*, SUTL, SYST, ZODI_FGR	
	RFC_TYPE	FUGR	
S_TCODE	TCD	SU53, SE38	
S_ADMI_FCD	S_ADMI_FCD		MENO
S_CTS_ADMI	CTS_ADMFCT	TABL	
S_DATASET	ACTVT	34	
	FILENAME	*	
	PROGRAM	SAPLSTRF	
S_TABU_DIS	ACTVT		03
	DICBERCLS		MA
S_DEVELOP	ACTVT	01, 02, 03, 06	07, 16
	DEVCLAS	\$TMP, ZODI_LKM_PCKG, ZODI_RKM_PCKG	
	OBJNAME	Z\$\$XRFC, ZODI_FGR, ZODI_FGR_PROD, ZODI_LKM_PCKG, ZODI_RKM_PCKG	
	OBJTYPE	DEVC, FUGR, PROG	
S_TRANSPRT	P_GROUP		*
	ACTVT	01, 02, 03, 06	06 (NR)
S_RS_ADMWB	TTYPE	DTRA, TASK	
	ACTVT	03, 16	16 (NR)
S_RS_ICUBE	RSADMWBOBJ	INFOBJECT	
	ACTVT	03	
	RSICUBEOBJ	DATA, DEFINITION	DEFINITION
	RSINFOAREA	NODESNOTCONNECTED	0*

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
S_RS_IOBJ	RSINFOCUBE	0*, Z*	1*
	ACTVT	03	
	RSIOBJ	0*, Z*	
	RSIOBJCAT	*	0*, 0CHANOTASSIGNED, Z*
S_RS_ODSO	RSIOBJPART	DEFINITION	DATA, DEFINITION
	ACTVT	03	
	RSINFOAREA	0*, NODESNOTCONNECTED	
	RSODSOBJ	0*, Z*	
	RSODSPART	DATA, DEFINITION	

Authorizations Required for RKM SAP BW Execution

The following authorizations are required for first time installation of RKM SAP BW.

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
S_RFC	ACTVT	16	
	RFC_NAME	RFC1, RS*, S*, SUTL, SYST, ZODI_FGR	
	RFC_TYPE	FUGR	
S_TCODE	TCD	SU53	
S_TABU_DIS	ACTVT	03	
	DICBERCLS	MA	SC
S_DEVELOP	ACTVT	*	NR
	DEVCLASS	*	NR
	OBJNAME	*	NR
	OBJTYPE	*	NR
	P_GROUP	*	NR
S_RS_ADMWB	ACTVT	03, 16	
	RSADMWBOBJ	INFOBJECT	
S_RS_ICUBE	ACTVT	03	
	RSICUBEBOBJ	DATA, DEFINITION	
	RSINFOAREA	NODESNOTCONNECTED	0*, NODESNOTCONNECTED (NR)
	RSINFOCUBE	0*, Z*	
S_RS_IOBJ	ACTVT	03	
	RSIOBJ	0*, Z*	Z* (NR)
	RSIOBJCAT	*	0*
	RSIOBJPART	DEFINITION	
S_RS_ODSO	ACTVT	03	

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
	RSINFOAREA	0*, NODESNOTCONNECTED	T*, Z*, NODESNOTCONNECTED (NR)
	RSODSOBJ	0*, Z*	*U
	RSODSPART	DATA, DEFINITION	

Authorizations Required for LKM SAP BW Upload (Background mode)

The following authorizations are required for running the mappings using SAP BW LKMs in a SAP development environment when LKM option `BACKGROUND_PROCESSING` is set to `true` (recommended default).

Note: Developer Key is required in this authorization.

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
S_RFC	ACTVT	16	
	RFC_NAME	RFC1, SDIFRUNTIME, SYST, ZODI_FGR, ZODI_FGR_PROD*	
	RFC_TYPE	FUGR	
S_TCODE	TCD	SE38, SU53	
S_BTCH_ADM	BTCADMIN	Y	
S_BTCH_JOB	JOBACTION	RELE	
	JOBGROUP	'	
S_CTS_ADMI	CTS_ADMFCT	TABL	
S_DATASET	ACTVT	06, 33, 34	
	FILENAME	*\\DEL-7TMK2BS\odi agentbox*	
	PROGRAM	SAPLSTRF, ZODI*	
S_DEVELOP	ACTVT	01	02, 03, 06, 16
	DEVCLASS		
	OBJNAME	ZODI_FGR_PROD*	
	OBJTYPE	FUGR	PROD
	P_GROUP		
S_PROGRAM	P_ACTION	NR	EDIT, SUBMIT, VARIANT
	P_GROUP	NR	
S_TRANSPRT	ACTVT	01, 02, 03	03 (NR)
	TTYPE	DTRA, TASK	
S_RS_AUTH	BIAUTH	0*	0*
S_RS_ICUBE	ACTVT	03, 06	NR

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
	RSICUBE OBJ	DATA, DEFINITION	NR
	RSINFOAREA	NODESNOTCONNECTED	NR
	RSINFOCUBE	0*, Z*	NR

Authorizations Required for LKM SAP BW Execution (Background mode)

The following authorizations are required for running the mappings using SAP BW LKMs in a SAP production environment when LKM option BACKGROUND_PROCESSING is set to true (recommended default).

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
S_RFC	ACTVT	16	
	RFC_NAME	RFC1, SDIFRUNTIME, SLST, SYST, ZODI*	
	RFC_TYPE	FUGR	
S_TCODE	TCD	SU53	
S_BTCH_ADM	BTCADMIN	Y	
S_BTCH_JOB	JOBACTION	RELE	
	JOBGROUP	' '	
S_DATASET	ACTVT	06, 33, 34	
	FILENAME	\\10.30.0.201\ODI_SAPTEST*	
	PROGRAM	ZODI*	
S_DEVELOP	ACTVT	NR	01, 02, 03, 06, 07, 16
	DEVCLASS	NR	
	OBJNAME	NR	S_TCODE
	OBJTYPE	NR	SUSO
	P_GROUP	NR	
S_RS_AUTH	BIAUTH	NR	0*
S_RS_ICUBE	ACTVT	03	
	RSICUBE OBJ	DATA, DEFINITION	
	RSINFOAREA	NODESNOTCONNECTED	0*, Z*, NODESNOTCONNECTED (NR)
	RSINFOCUBE	0*, Z*	

Authorizations Required for LKM SAP BW Upload (Dialog mode)

The following authorizations are required for running the mappings using the SAP BW LKMs in a SAP development environment when LKM option BACKGROUND_PROCESSING is set to false (non-default).

Note: Developer Key is required in this authorization.

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
S_RFC	ACTVT	16	
	RFC_NAME	RFC1, SDIFRUNTIME, SYST, ZODI_FGR, ZODI_FGR_PROD*	
	RFC_TYPE	FUGR	
S_TCODE	TCOD	SE38, SU53	
S_CTS_ADAMI	CTS_ADMFCT	TABL	
S_DATASET	ACTVT	06, 33, 34	
	FILENAME	*\\DEL-7TMK2BS\odi agentbox*,	
	PROGRAM	SAPLSLOG*, SAPLSTRF*, SAPLZODI*	
S_DEVELOP	ACTVT	01, 02, 03, 06	
	DEVCLASS		
	OBJNAME	ZODI_FGR_PROD*	
	OBJTYPE	FUGR, PROG	
	P_GROUP		
S_PROGRAM	P_ACTION	NR	EDIT, SUBMIT, VARIANT
	P_GROUP	NR	
S_TRANSPRT	ACTVT	01	
	TTYPE	DTRA, TASK	
S_RS_AUTH	BIAUTH	NR	0*
S_RS_ICUBE	ACTVT	03	NR
	RSICUBEOBJ	DATA, DEFINITION	NR
	RSINFOAREA	NODESNOTCONNECTED	NR
	RSINFOCUBE	0*, Z*	NR

Authorizations Required for LKM SAP BW Execution (Dialog mode)

The following authorizations are required for running the mappings using the SAP BW LKMs in a SAP production environment when LKM option `BACKGROUND_PROCESSING` is set to `false` (non-default).

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
S_RFC	ACTVT	16	
	RFC_NAME	RFC1, SDIFRUNTIME, SYST, ZODI_FGR, ZODI_FGR_PROD*	
	RFC_TYPE	FUGR	
S_TCODE	TCOD	SU53	
S_DATASET	ACTVT	06, 33, 34	

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
	FILENAME	*\\DEL-7TMK2BS\odi agentbox*,	
	PROGRAM	SAPLSLOG*, SAPLSTRF*, SAPLZODI*	
S_RS_AUTH	BIAUTH	NR	0*
S_RS_ICUBE	ACTVT	03	
	RSICUBE0B	DATA, DEFINITION	DATA (NR)
	RSINFOAREA	NODESNOTCONNECTED	NR
	RSINFOCUBE	0*, Z*	NR
S_RS_IOBJ	ACTVT	NR	03
	RSIOBJ	NR	
	RSIOBJCAT	NR	
	RSIOBJPART	NR	DEFINITION
S_RS_ODSO	ACTVT	NR	03
	RSINFOAREA	NR	
	RSODSOBJ	NR	
	RSODSPART	NR	DEFINITION

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 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 16:07:27 200
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 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 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:
SYSINR: 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.

Uninstalling ODI SAP Components

This appendix describes how to uninstall ODI components from your SAP system.

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.

This appendix includes the following sections:

- [Updating ODI SAP Components](#)
- [Uninstalling ODI SAP Components](#)

Updating ODI SAP Components

During first-time installation the RKM installs some ODI objects into the SAP system. This installation consists of two parts: some base objects and some RFCs. When the RKM options `UPLOAD_ABAP_BASE` and `UPLOAD_ABAP_CODE` are both set to `true`, the base objects and the RFCs are installed. Such full installation requires that no ODI SAP objects are installed in the SAP system. See section [Uninstalling ODI SAP Components](#) if needed.

If the ODI objects installed into your SAP systems have been installed by RKM SAP ERP v32 or later / RKM SAP BW v23 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. For older installations the package name is `ZODIBW_PCKG`.
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 one by one and then delete the development class.
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. This step applies only to the installations done by RKM SAP ERP v32 or later.