Oracle® Data Integrator

Getting Started with SAP ABAP BW Adapter

10g Release 3 (10.1.3)

November 2009

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

Table of Contents

Introduction

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 Data Integrator - Getting Started with an ETL Project*.

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

Overview

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

- Reads data from SAP BW system
- Loads this data into 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 data stores (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)** connects to SAP BW and retrieves data from the SAP BW data targets defined in the SAP Model. It is used in the interfaces. The LKM is in charge of loading source data from a SAP BW System to an Oracle staging area.

Setting up the Environment

Before You Begin

System Requirements and Certification

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

The Oracle Data Integrator requirements are listed in the *Oracle Data Integrator Installation Guide* available from the Oracle Technology Network (OTN).

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

- The minimum patch level for Oracle Data Integrator 10gR3 is version 10.1.3.5.1_01 (Metalink Patch #8909138) or above.
- A JCo version compatible with adapter must be used. The list of supported JCo versions is available in the Compatibility Matrix available from the Oracle Technology Network (OTN). A minimum version of JCo 3.0.2 is required.
- A JVM version compatible with both Oracle Data Integrator and JCo must be used. A minimum version of JVM 1.5 is required due to JCo pre-requisites.
- 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 interface (when this agent runs on the FTP server machine), or remotely (when this agent is located on a different machine than the FTP server). This FTP server must be accessible over the network from both the SAP BW machine and the agent machine.
- 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 BW Adapter requires privileges to perform set up and execution of programs. Please provide your administrators with the list of privileges listed *in Appendix A – SAP 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

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 in 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.

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 FTP Setup

This section describes how to verify the FTP setup and 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 SAPFTPA 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 SAPFTPA destination.
6. Click **Test connection**.

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

Step 2: Testing FTP connection

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

   This should report a successful connection. Similar to this 250 CWD successful. "/" is current directory. If not, 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.

Please find the steps below 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.

Installing Oracle Data Integrator

Before starting with this project, you need to install and configure Oracle Data Integrator. Refer to the guide *Oracle Data Integrator - Getting Started with an ETL Project* and the *Oracle Data Integrator Installation Guide* for more information.

Installing and Configuring the Oracle DB Target

This project uses an Oracle database engine as the target and as the staging area for the integration interfaces. You can download for example an Oracle XE database from the Oracle Technology Network (OTN). Install and configure this database.

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

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

-- Create demo target 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 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);


-- 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 JCo to connect to the SAP system. JCo must be configured before proceeding with the project.

To install and configure JCo:

1. Download a supported JCo version for your configuration from http://service.sap.com/connectors. Check the supported JCo version in the Compatibility Matrix available at Oracle Technology Network. Note that a minimum version of JCo 3.0.2 is required.
1. Unzip the appropriate distribution package into an arbitrary directory `{sapjco-install-path}`.
2. Follow the installation instructions in `{sapjco-install-path}/javadoc/installation.html` for the respective platform.
3. Copy `sapjco3.jar` and `sapjco3.dll` (or respective binary) into the `oracledi/drivers` directory.
4. Check the JCo installation as described in the *Validating the SAP Connection* section.

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

Setting up an FTP Server

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

Configuring Oracle Data Integrator

Set up the Topology

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

1. Connect to Topology Manager.
2. If this SAP ABAP technology does not exist in your Master Repository, import the SAP ABAP technology in Synonym INSERT_UPDATE mode from the `/impexp` folder.

3. Perform an upgrade of the Master Repository. Refer to the Oracle Data Integrator Installation Guide on OTN for more information on the Master Repository upgrade process.
4. In Topology Manager, open the JavaBeanShell technology and check on the Language tab that the JYTHON language is listed. If not, add it.
5. Create a File data server pointing to an existing FTP server into which the extraction file will be pushed from SAP and picked up for SQL*Loader. Set the parameters for this data server as follows:
   - **Host (Data Server)**: FTP server IP host name or IP address.
   - **User**: Username to log into FTP server.
   - **Password**: Password for the user.
6. In this File data server create a Physical Schema representing the folder in the FTP host where the extraction file will be pushed. Specify the Data and Work Schemas as follows:
   - **Data Schema**: Path on the FTP server to upload or download extraction files from the remote location. This path is used when uploading extraction files from the SAP 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) is set to NONE. As Work Schema is an OS file name, slashes/ backslashes should be used according to OS. Path names need to end on slash/ backslash.

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

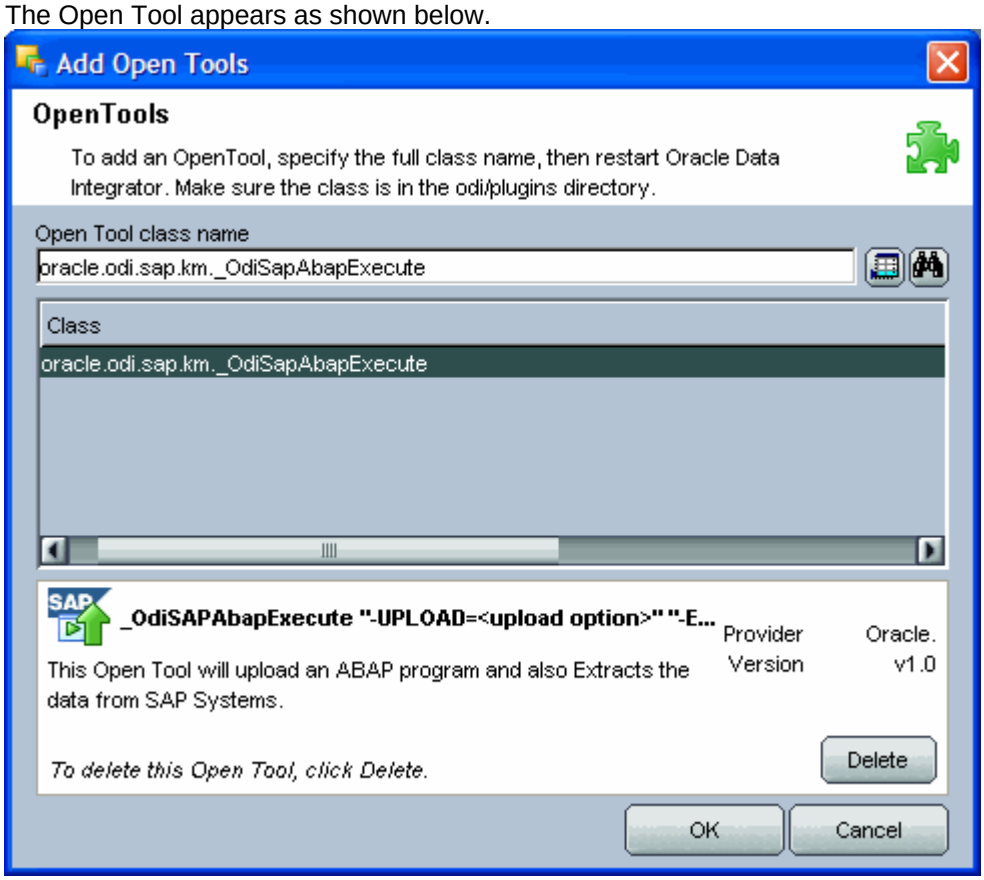
   Refer to the *File Transfer Configurations* section in the *SAP ABAP* chapter of the *Oracle Data Integrator Knowledge Modules Reference Guide*.
7. If the corresponding Logical Schema called `File Server for SAP ABAP` does not exist yet, create it. This Logical Schema name is fixed and must be mapped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

Add the Open Tool

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

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

5. The Open Tool appears as shown below.



6. Click **OK**.

Configuring the Topology

This project 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 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.

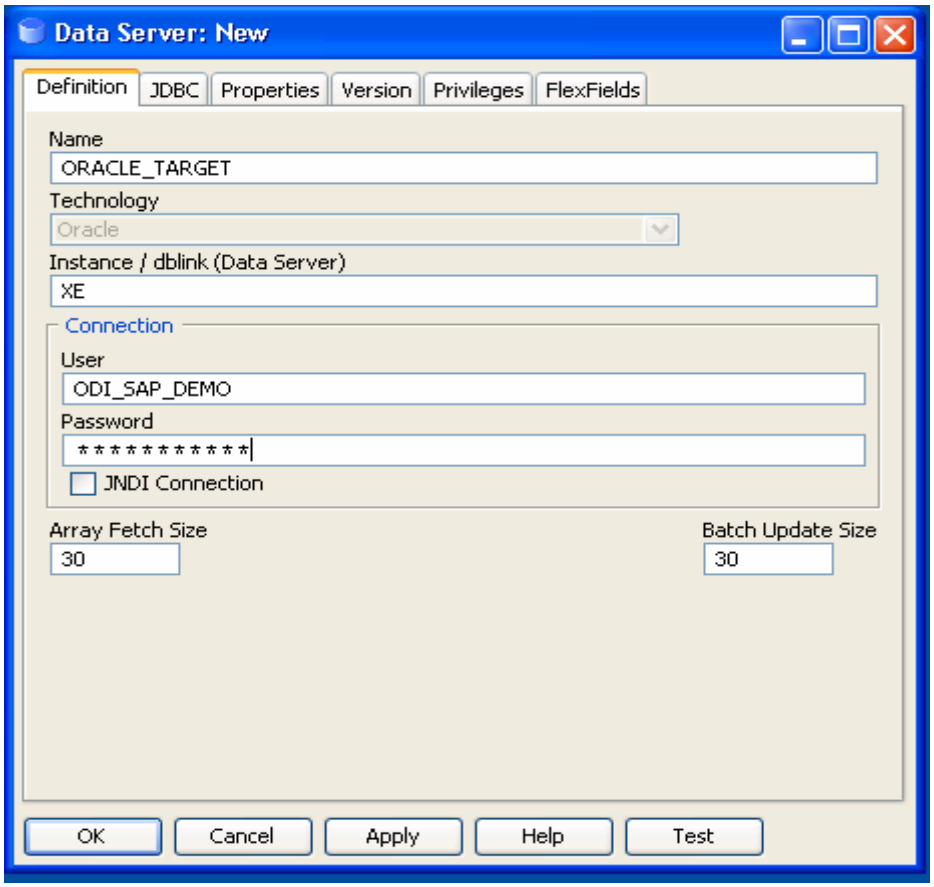
- **Delta Extraction:**

  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.

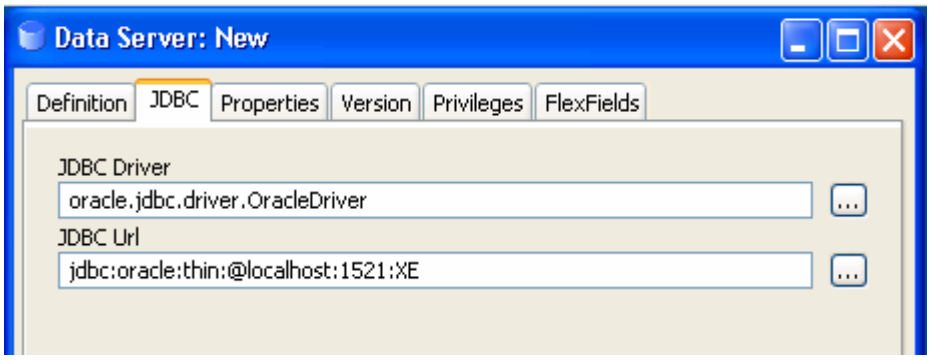
Configuring the Oracle Data Server

1. Connect to Topology Manager.
2. In the Physical Architecture tree view, expand the Technologies node and select the `Oracle` technology.

3. Right-click and select **Insert Data Server**.
4. Enter the data server Definition as shown below:



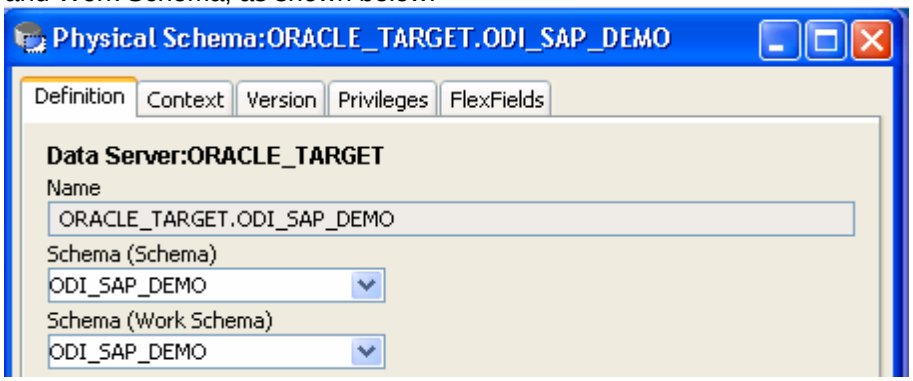
5. Select the JDBC tab, and enter the JDBC connection information to your data server as shown below:



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

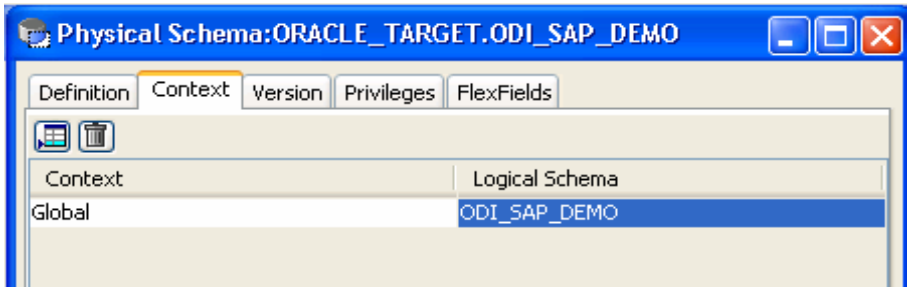
6. Click **Test** to test the connection to this data server.
7. Click **OK** to save your data server.

8. In the Physical Schema windows that appears, select the `ODI_SAP_DEMO` for both Schema and Work Schema, as shown below:



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

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



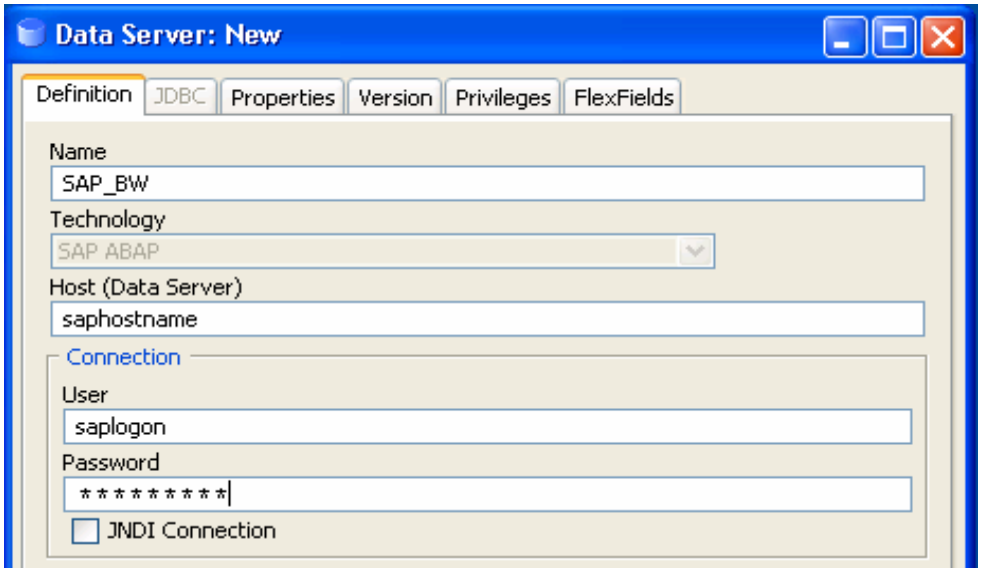
11. Click **OK** to save.

Configuring the SAP Source Server

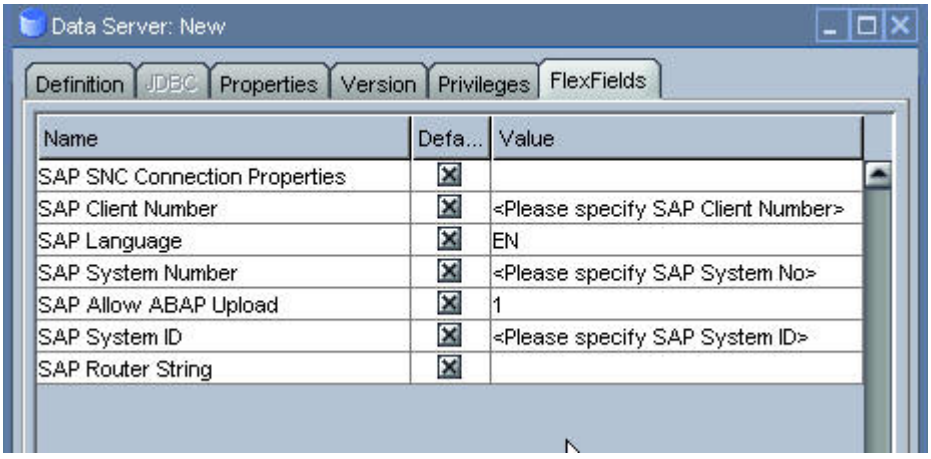
Configuring the Data Server

1. In the Physical Architecture tree view, expand the Technologies node, select the SAP ABAP technology.
2. Right-click and select **Insert Data Server**.
3. Enter the data server definition. Set the parameters for this data server as follows:
   - **Name**: SAP_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.

2. Click **OK**.

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

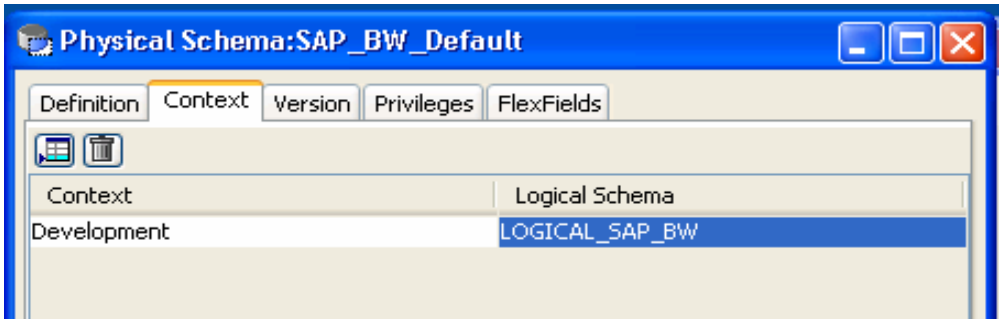
Except for the data server name, all the parameters that you provide while defining the SAP Data server should be provided by the SAP Administrators. See *Getting the Right Privileges* for more information.

The SAP BW Adapter requires privileges to perform set up and execution of programs. Please provide your administrators with the list of privileges listed in *Appendix A – SAP BW Required Privileges*. These privileges are required for the SAP BW user, which will be used by ODI to connect SAP BW system.

See *Gathering SAP Connection Information* for more information about the SAP connection 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. Click **OK**.

Create a New Project

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

To create a new Project for SAP BW:

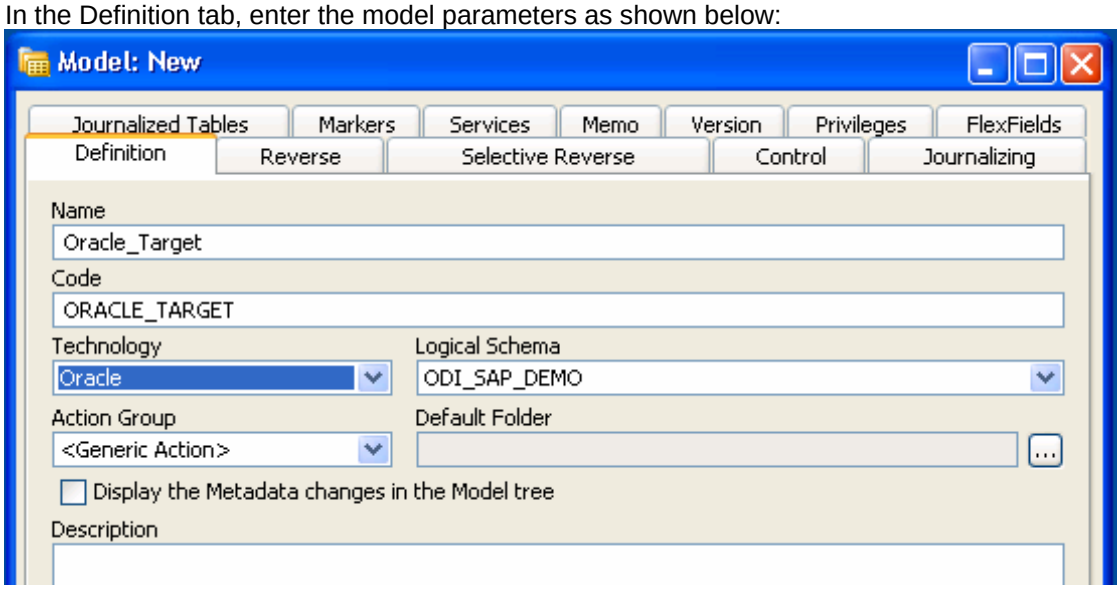
1. Connect to Designer.
2. In the Projects tree view click **Insert Project**.
3. In the Definition tab, enter `SAP BW Demo` for the Project name.
4. Click **OK** to save your Project.
5. Expand the `SAP Demo` Project, select the `First Folder` node.
6. Rename this folder to `SAP_BW_DataTargets`.
7.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

Reverse-Engineering the Data Models

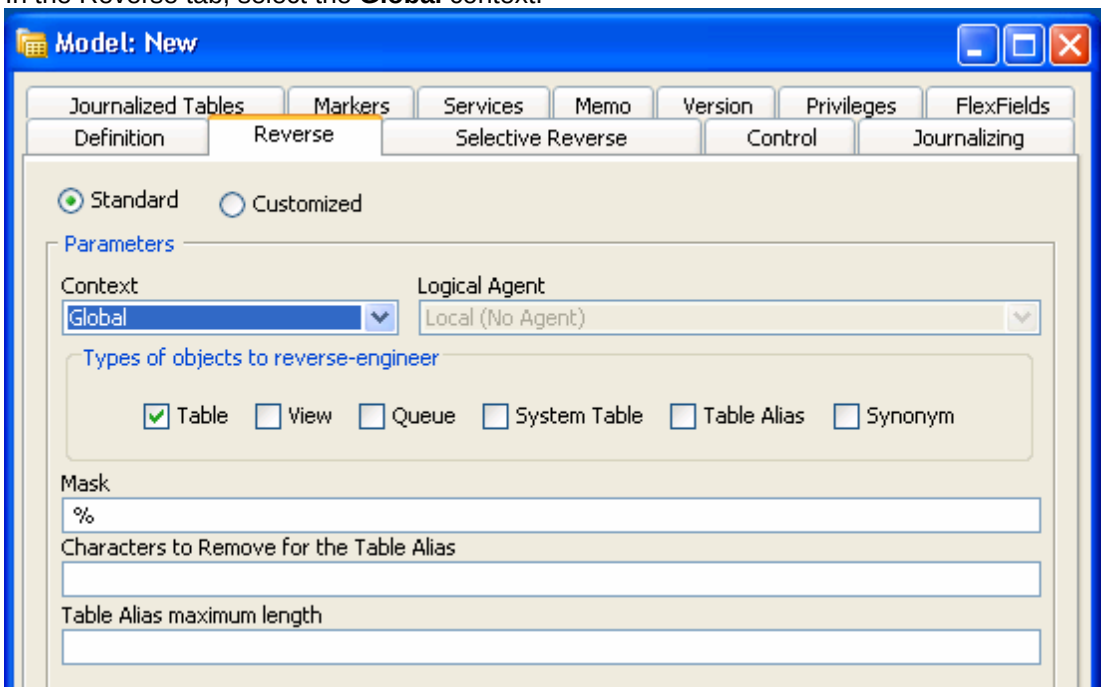
Reverse-Engineering the Oracle Target

To reverse-engineer the Oracle data model:

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

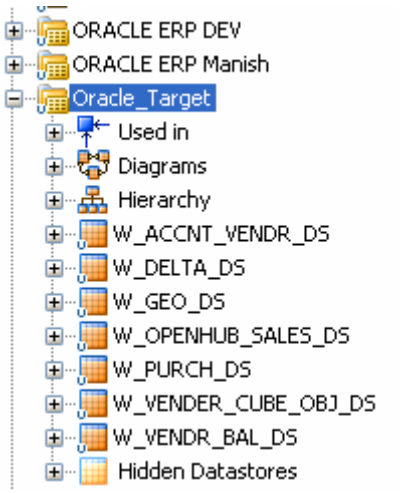


4. In the Reverse tab, select the **Global** context.



5. Click **Reverse**, then **OK** to save your model.

6. The `W_PURCH_DS, W_ACCNT_VENDR_DS, W_VENDR_BAL_DS, W_VENDER_CUBE_OBJ_DS, W_OPENHUB_SALES_DS, W_DELTA_DS` datastore appears in your Oracle model, as shown below.



Reverse-Engineering the SAP BW Source Data Targets

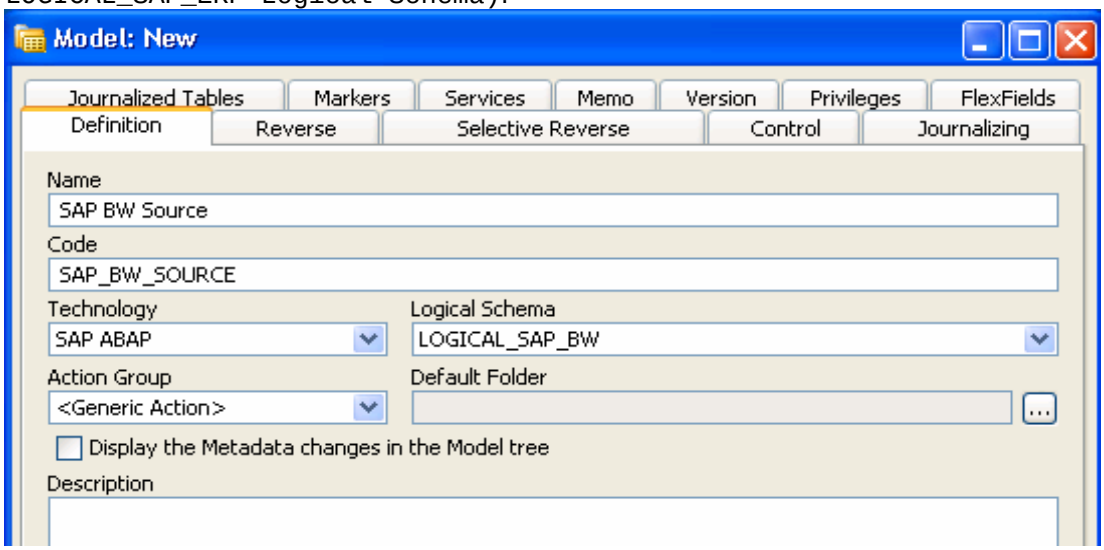
To reverse-engineer the SAP BW Source data targets you need to perform the following tasks:

1. Create the Data Model
2. Validate the SAP BW Connection
3. Start the Reverse-Engineering Process

Creating the Data Model

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

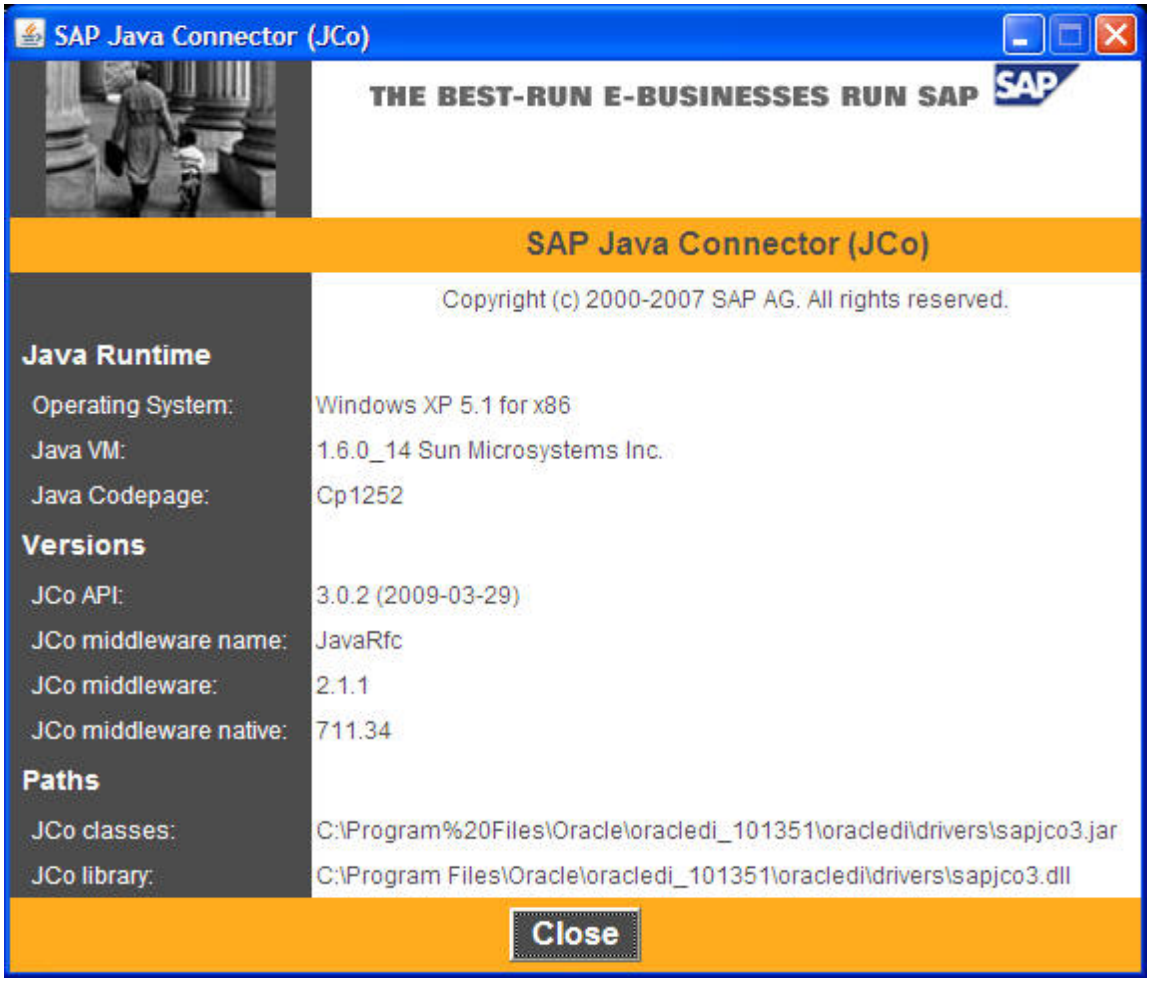
LOGICAL_SAP_ERP Logical Schema).



3. In the Reverse tab:
   a. Select the `Global` context.
   b. Select `Customized` option.
   c. 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** 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 the Pre-requisites section for more details.) and verify that a path for the JCo Library is given (`.dll` for Windows).
If this is not the case or you see any error message in this about dialog, please review the SAP JCo installation, as described in your SAP JCo package and run this test again.

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

4. Once you have validated SAP JCo installation, open ODI Operator.
5. In the Operator, expand the Reverse-Engineering session down to the task level.
6. Verify whether the task “Test SAP Connection” has been executed successfully.
   If not, view the task details to identify the connection problem.

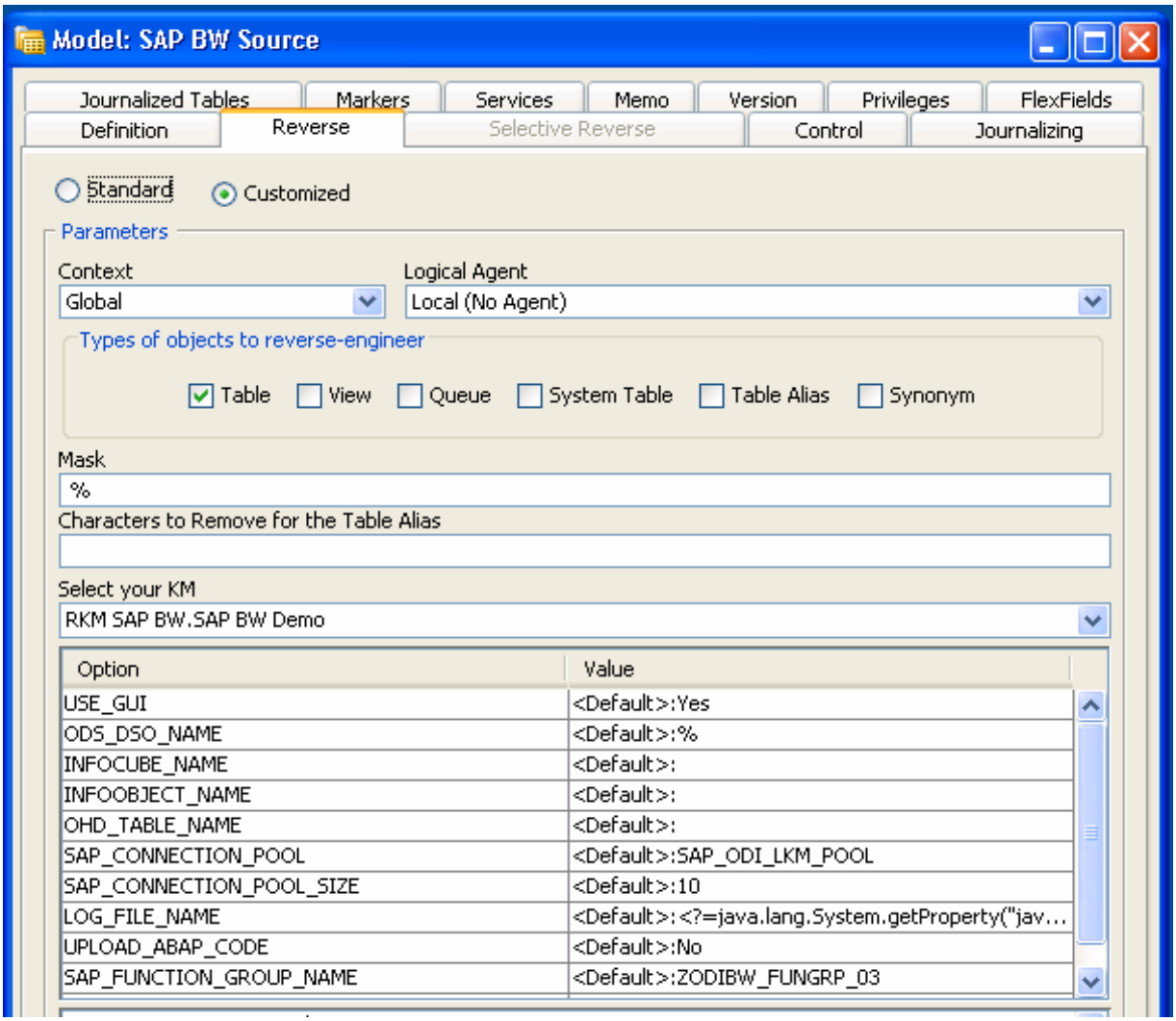
   Do not continue until you have successfully validated the SAP BW connection!

Note: If you want to validate the SAP BW connectivity independently 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 data stores:

1. In the Models tree view, open the `SAP BW Source` Model.
2. In the Reverse tab:
    a. Select the `Global` context.
    b. Select the `Customized` option.
    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. Set other parameters according to your SAP configuration. Refer to the *Oracle Data Integrator Knowledge Modules Reference Guide* for more information on the RKM options.



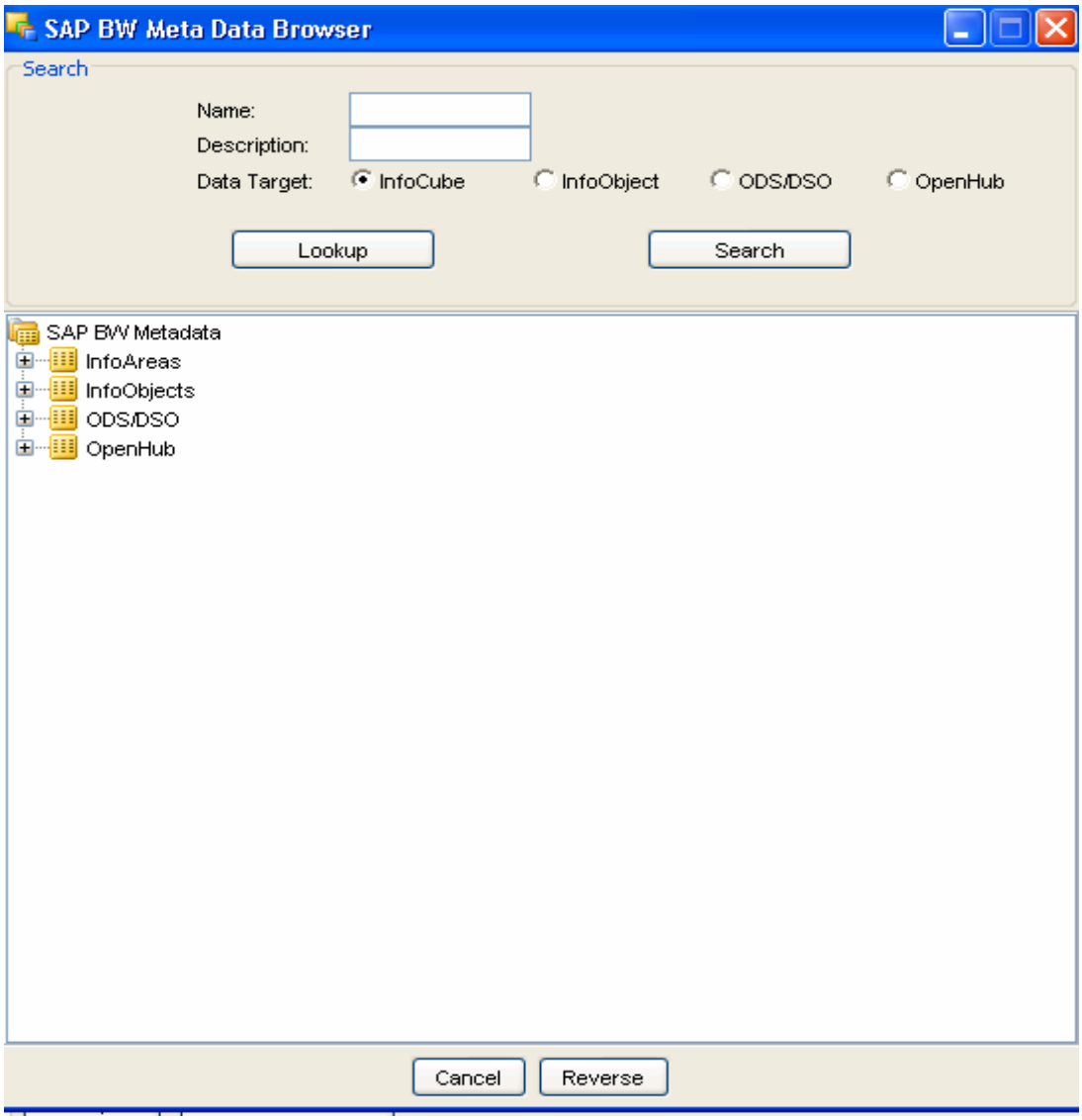
3. Click **Apply** to save your changes.
4. Click **Reverse** to start the reverse-engineering process.
5. Click **OK.**
6. The Sessions Started Dialog is displayed.
7. Click **OK**.

Using the SAP BW Metadata Browser

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

> **Note**: In case that the below dialog does not show up, see ODI Operator for details and read the log `files <System Temp Dir>/sap_rkm_bw_<ODI Session Number>.log`.

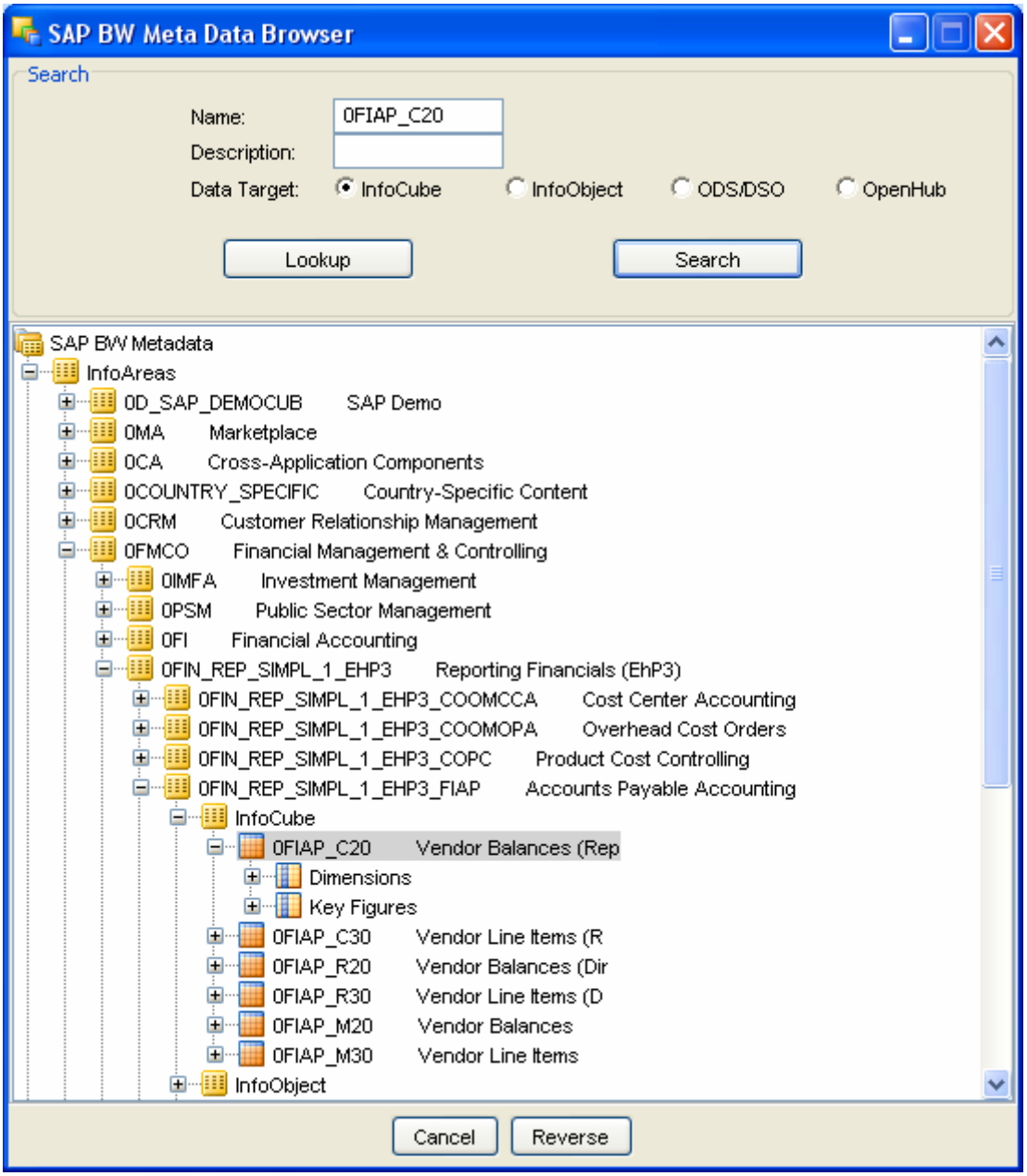
Refer to the *Log Files* section in the *SAP ABAP – BW* chapter of the *Oracle Data Integrator Knowledge Modules Reference* Guide for more details.



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.

Reverse-Engineering without GUI

If you want to perform a reverse-engineering without using the SAP BW Metadata Browser, you can 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 *Oracle Data Integrator Knowledge Modules Reference Guide* for more information on the RKM options.

Note: It is not possible to view the data in the SAP 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 option is selected (this option is typically activated only once). The ABAP programs are generated in the SAP function group given in the SAP_FUNCTION_GROUP_NAME RKM option.

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.

Create the Integration Interface

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

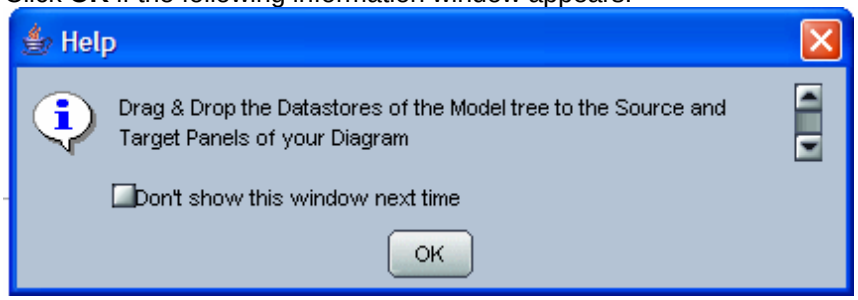
- The Info Cube `0FIAP_C20` data target to `W_VENDR_BAL_DS` Oracle table.
- The Info Object `0CREDITOR` data target to `W_ACCNT_VENDR_DS` Oracle table.
- The ODS/DSO `0BBP_PO` data target to `W_PURCH_DS` Oracle table.
- The Info Cube `0FIAP_C20` and Info Object `0CREDITOR` data targets join to `W_VENDER_CUBE_OBJ_DS` Oracle table.
- The Open Hub destination `ZOHD_SALE (/BIC/OHZOHD_SALE)` to `W_OPENHUB_SALES_DS` Oracle table.
- Delta Extraction can be done on any above said interface except Info Object. How to perform delta extraction is described in the Creating a Package for Delta Extraction section.

Info Cube Extraction

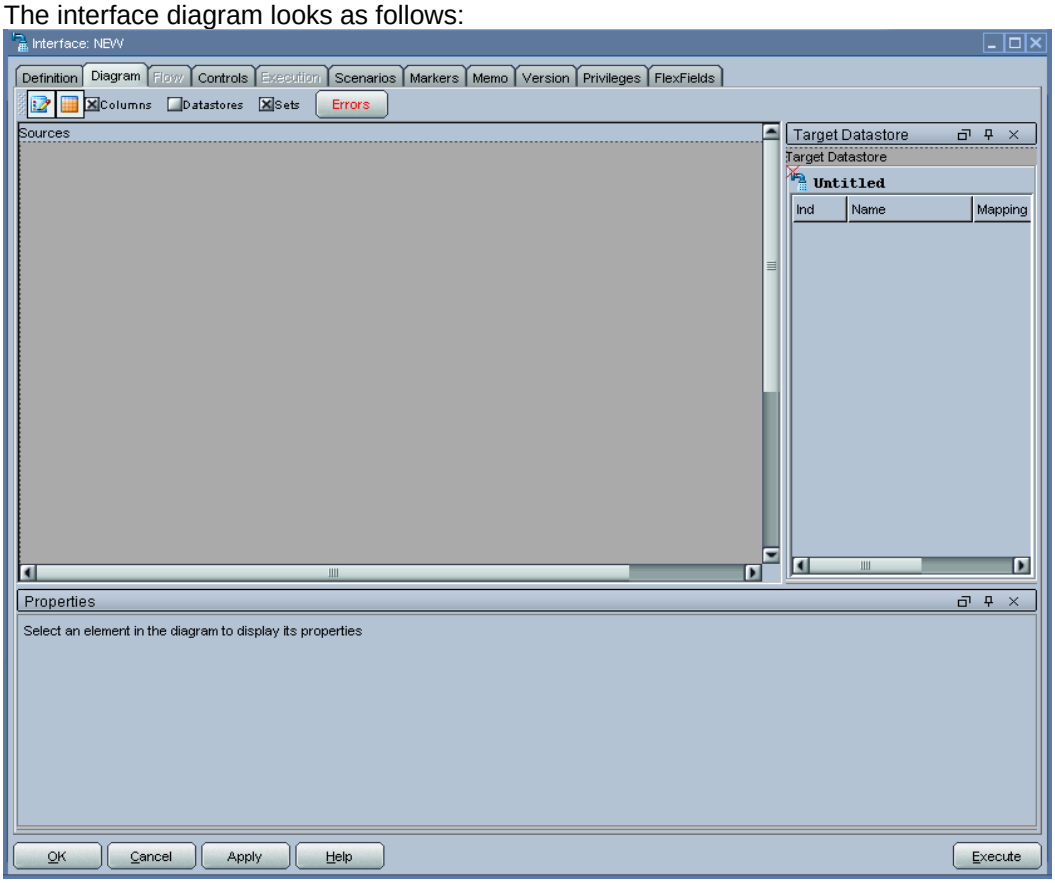
This section describes how to create an integration interface in ODI in order to load data from the SAP BW source model representing the InfoCube `0FIAP_C20` data target to the `W_VENDR_BAL_DS` Oracle target model.

Create the Interface

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 Interfaces node.
5. Right-click and select **Insert Interface**.
6. In the Interface Definition tab, enter the interface name: `InfoCube_SAP_BW_VendorBal.W_VENDR_BAL_DS`.
7. Select the Diagram tab.
8. Click **OK** if the following information window appears.



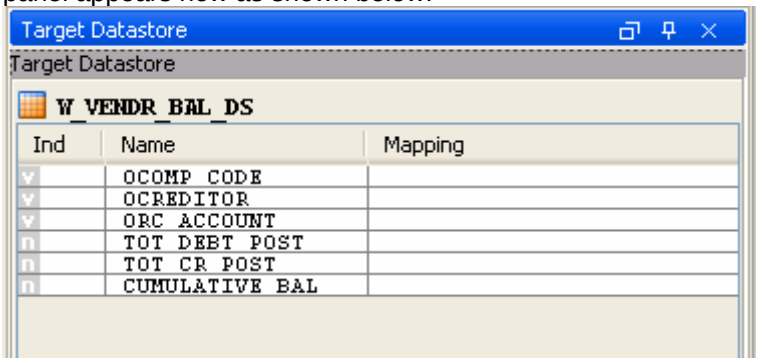
The interface diagram looks as follows:



Define the Source and Target Datastores

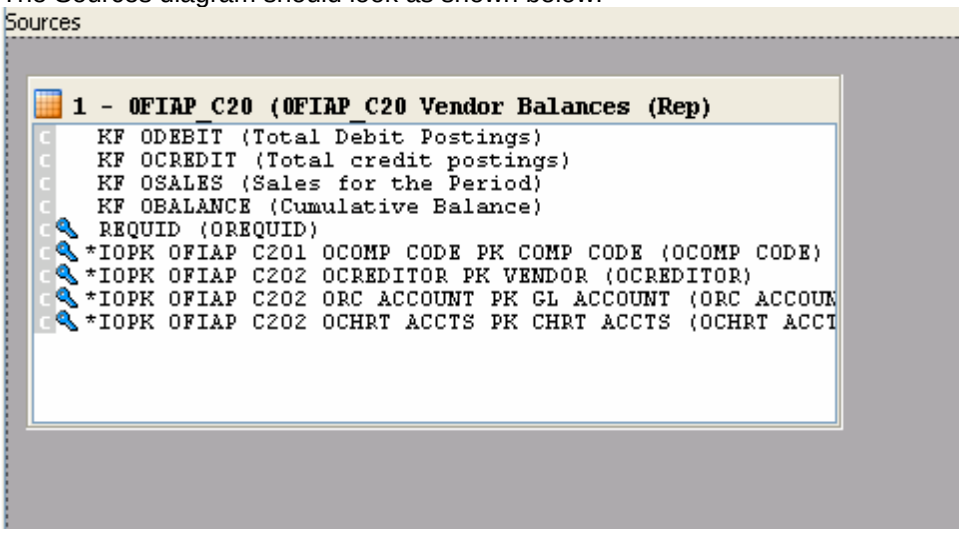
To define the source and target datastores:

1. In the Models tree view, select the `W_VENDR_BAL_DS` datastore from the `Oracle_Target` Model.
2. Drag this datastore into the Target Datastore panel (right area of the Diagram tab). This panel appears now as shown below:



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` diagram panel (left area of the diagram tab).
5. If a popup window prompting “Do you want to perform Automatic Mapping?” appears, click **No**.

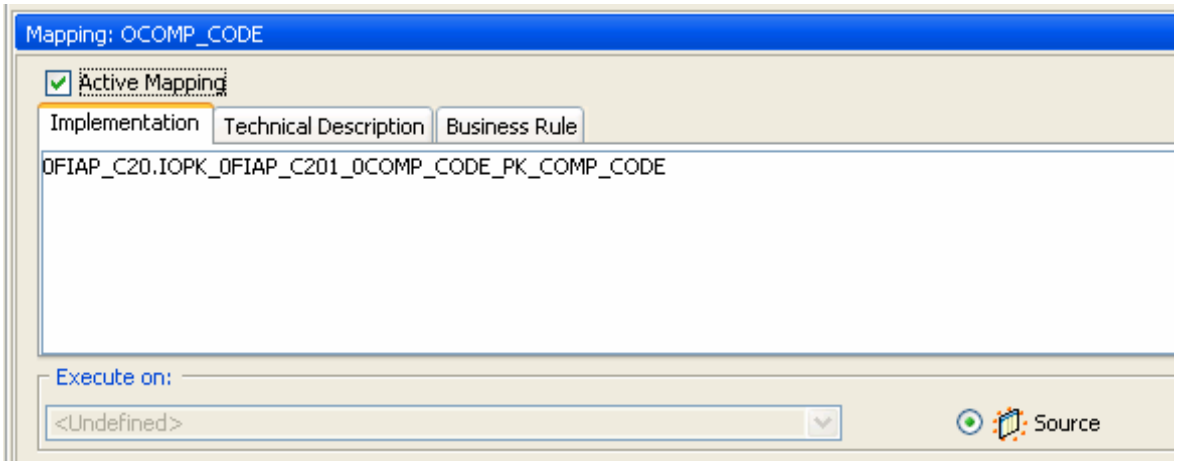
6. The Sources diagram should look as shown below:



Create the Mappings

To create the mappings of the target datastore:

1. Select the `0COMP_CODE` column from the target datastore.
2. Drag the `IOPK_0FIAP_C201_0COMP_CODE_PK_COMP_CODE` column from the `0FIAP_C20 Vendor Balances` source data target into the Implementation field of the Properties panel.
3. Make sure that the **Execution On** is set to **Source**. The mapping should look as shown below:

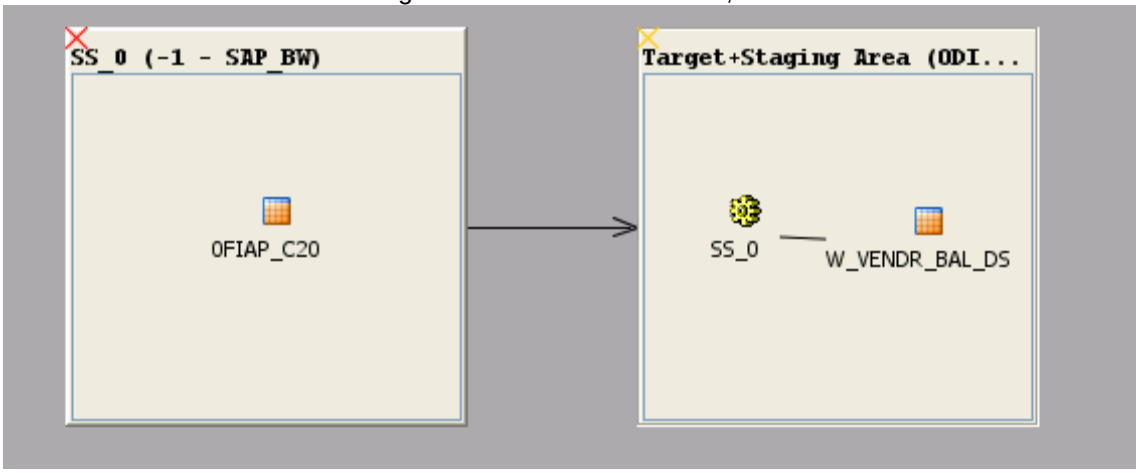


4. Repeat this operation to perform the following simple mappings:
   - **`W_VENDR_BAL_DS.OCREDITOR = 0FIAP_C20.IOPK_0FIAP_C202_0CREDITOR_PK_VENDOR`**
   - **`W_VENDR_BAL_DS.ORC ACCOUNT = 0FIAP_C20.IOPK_0FIAP_C202_0RC_ACCOUNT_PK_GL_ACCOUNT`**
   - **`W_VENDR_BAL_DS.TOTAL DEBIT POSTING = 0FIAP_C20.KF_0DEBIT`**
   - **`W_VENDR_BAL_DS.TOTAL CREDIT POSTING - 0FIAP_C20.KF_0DEBIT`**
   - **`W_VENDR_BAL_DS.CUMULATIVE BALANCE - 0FIAP_C20.KF_0BALANCE`**

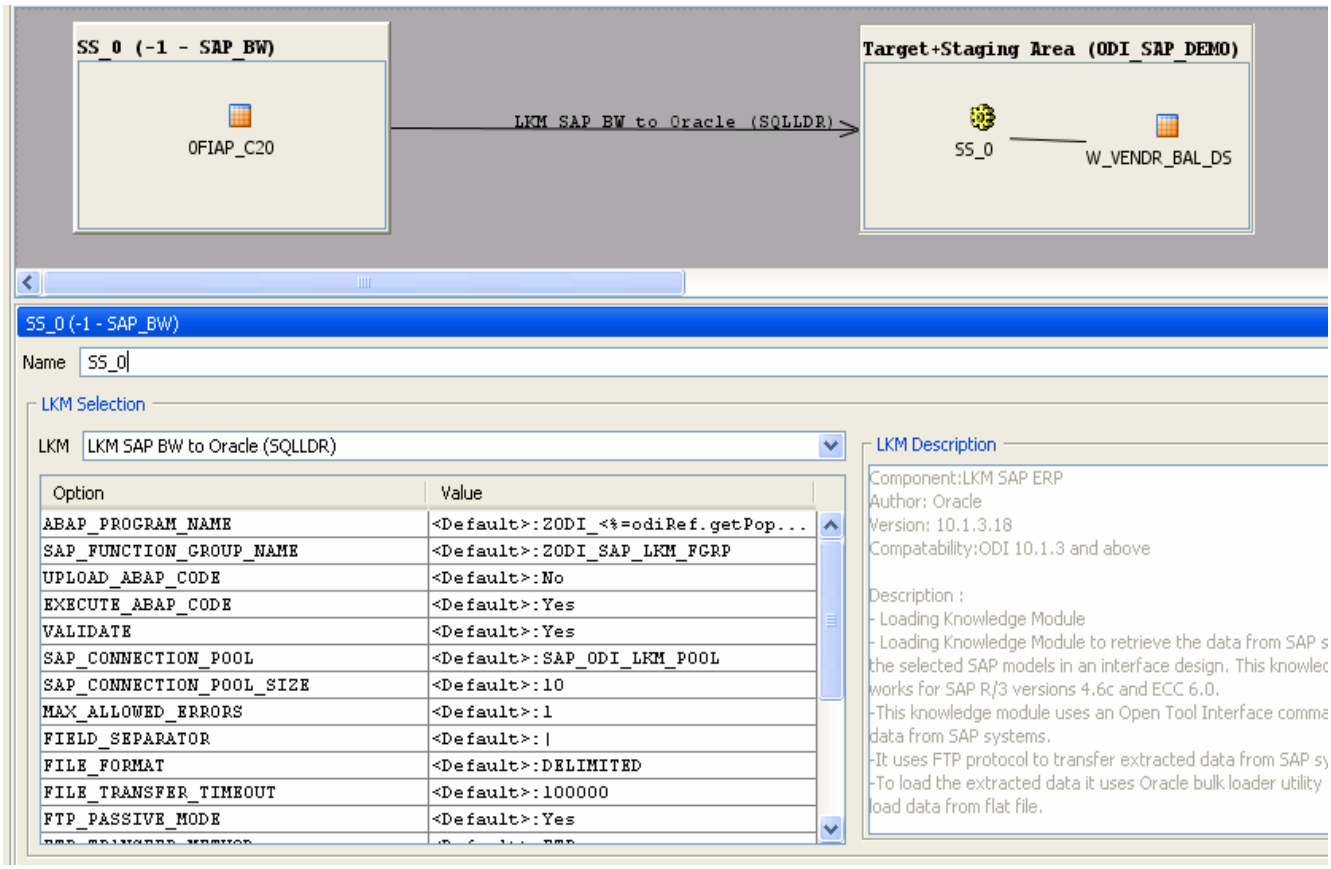
Define the Interface Flow

To define the interface flow:

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



2. Select the source set `SS_0 (SAP_BW)` that contains Info Cube `0FIAP_C20` data targets. This source set represents the source dataset made up of `SAP BW` data target.
3. In the LKM Selection box, select `LKM SAP BW to Oracle (SQLLDR)`.
4. Select the **Target + Staging Area** box. This represents the target Oracle database, which is also used as the staging area for this interface.
5. In the LKM Selection, select `LKM SAP BW to Oracle (SQLLDR)`.
6. In the IKM Selection, select `IKM Oracle Incremental Update`.
7. Set the IKM options as follows:
    a. Select `No` for the *FLOW_CONTROL* option.
    b. Leave other options as it is. The flow should look as shown below:



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

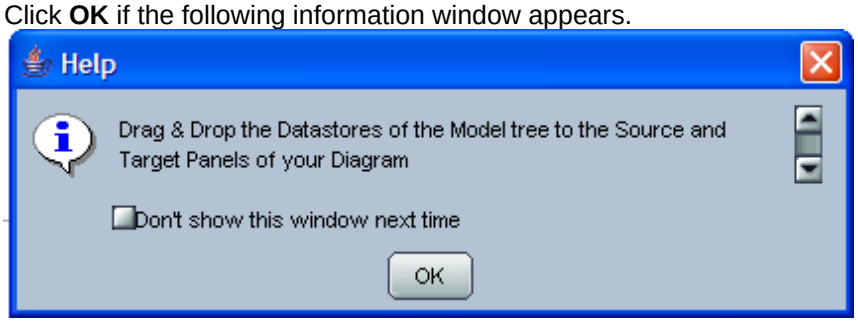
Info Object Extraction

This section describes how to create an integration interface 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.

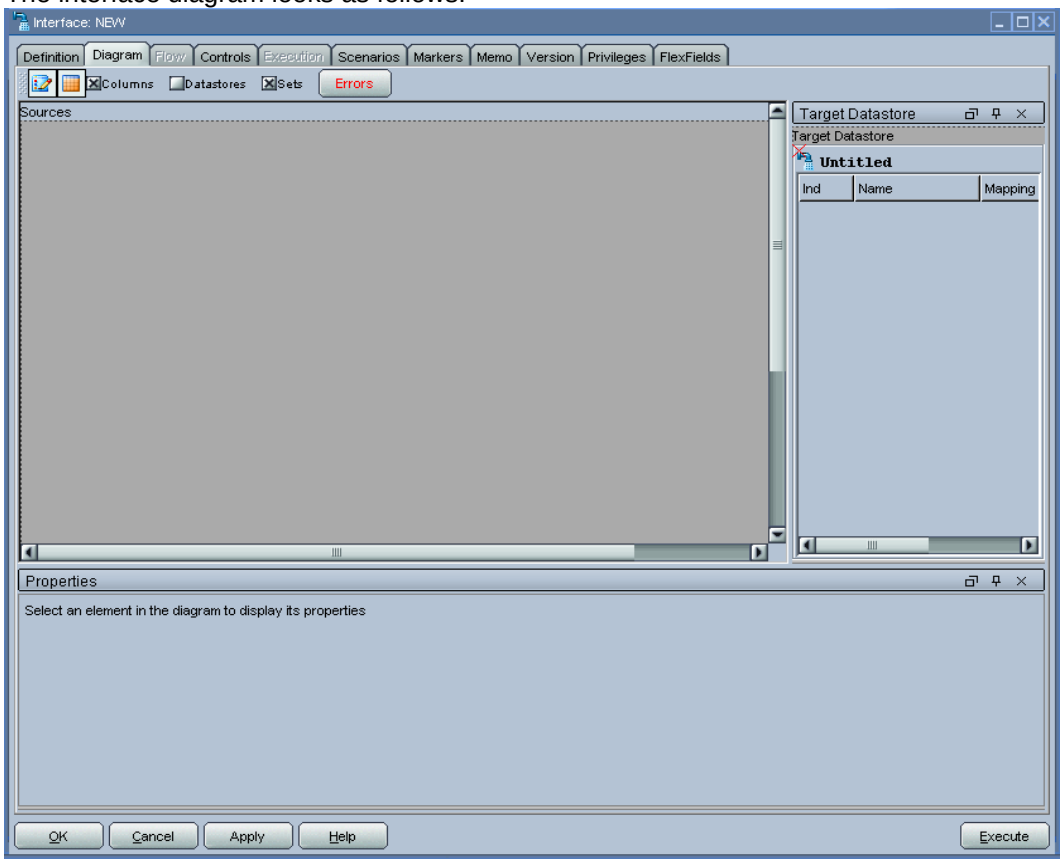
Create the Interface

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 Interfaces node.
5. Right-click and select **Insert Interface**.
6. In the Interface Definition tab, enter the interface name: `InfoObject_SAP_BW_VendorAcc.W_ACCNT_VENDR_DS`.
7. Select the Diagram tab.

8. Click **OK** if the following information window appears.



The interface diagram looks as follows:

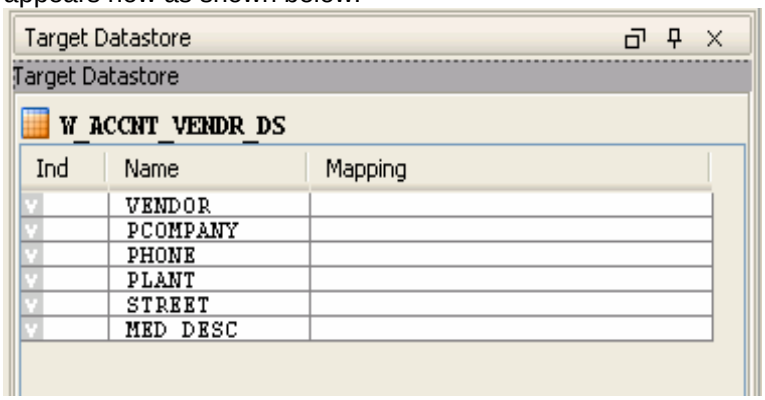


Define the Source and Target Datastores

To define the source and target datastores:

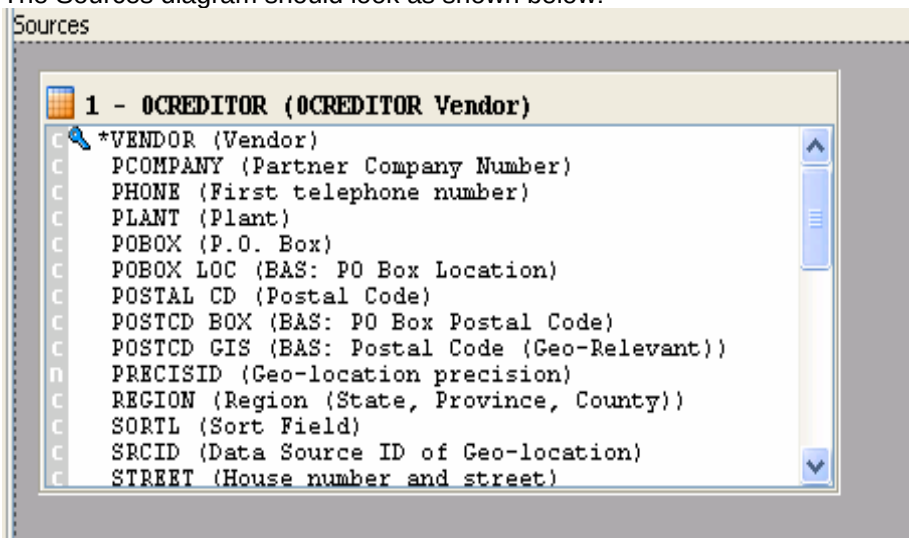
1. In the Models tree view, select the `W_ACCNT_VENDR_DS` datastore from the `Oracle_Target` Model.

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



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` diagram panel (left area of the diagram tab).
5. If a popup window prompting "Do you want to perform Automatic Mapping?" appears, click **No**.
6. Perform the same operation for the Info Object text table `TXT_0CREDITOR Texts: Char. Vendor` datastore from the `SAP BW Source` model.

   The Sources diagram should look as shown below:

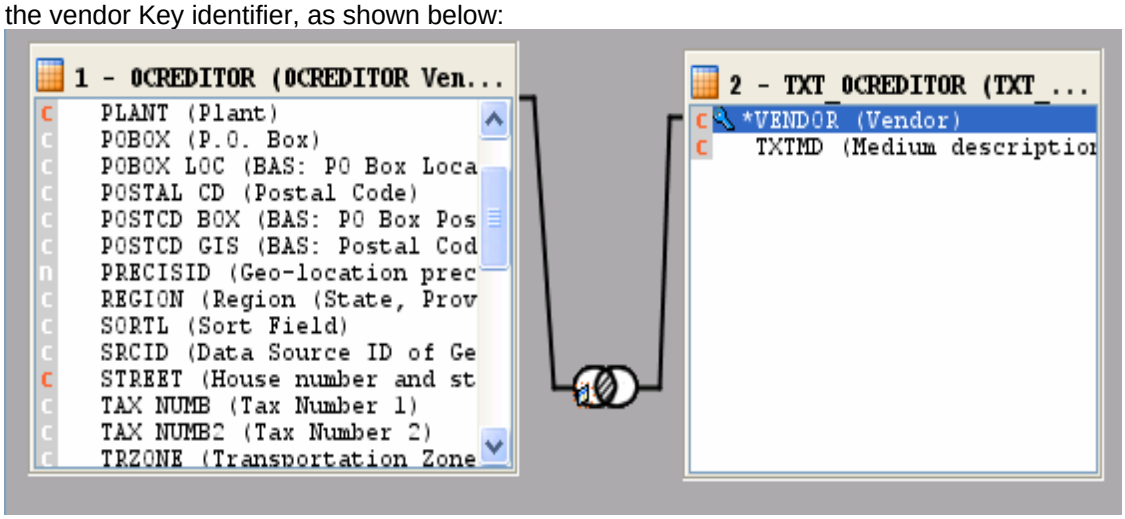


Define Joins between Info Object's Main table and Text Table

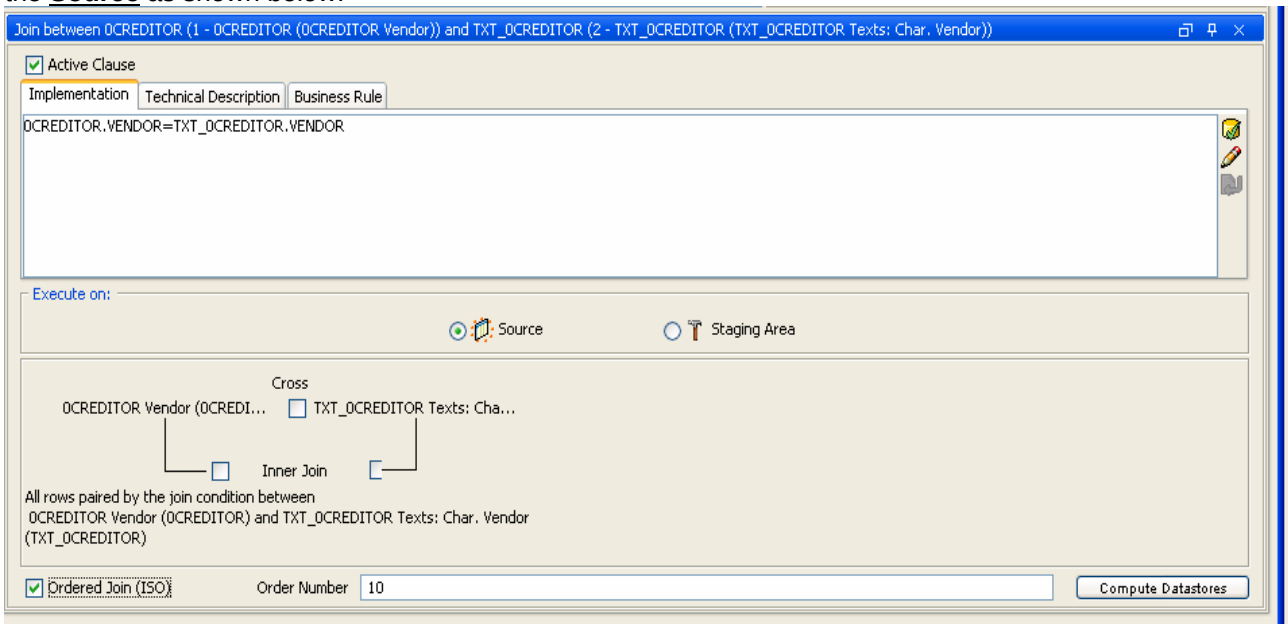
To create joins between the source datastores of Info Object's main table and text table in the interface:

1. In the Sources diagram when you drag the Info Object's text table `TXT_0CREDITOR Texts: Char. Vendor` the join will be automatically created on column `vendor` from the `0CREDITOR Vendor` Info object's main table datastore and the `Vendor` column of Info Object's text table `TXT_0CREDITOR Texts: Char. Vendor`. This defines the first join on

the vendor Key identifier, as shown below:



2. In the Properties panel, select `Ordered join` and make sure that this join is executed on the **Source** as shown below:



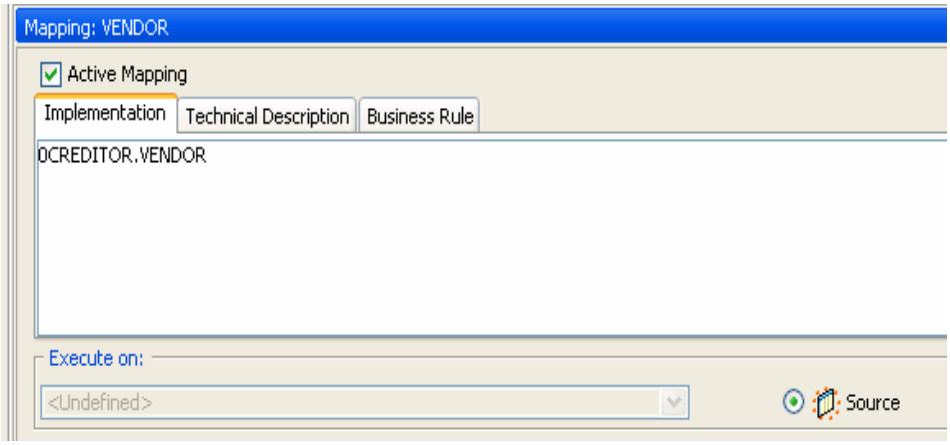
3. Make sure that table number of main table `0CREDITOR` is smaller than that of text table `TXT_0CREDITOR`. In *1:n relationships* the parent table has to be the first in a join.

Create the Mappings

To create the mappings of the target datastore:

1. Select the `0COMP_CODE` column from the target datastore.
2. Drag the `IOPK_0FIAP_C201_0COMP_CODE_PK_COMP_CODE` column from the `0FIAP_C20 Vendor Balances` source data target into the Implementation field of the Properties panel.

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

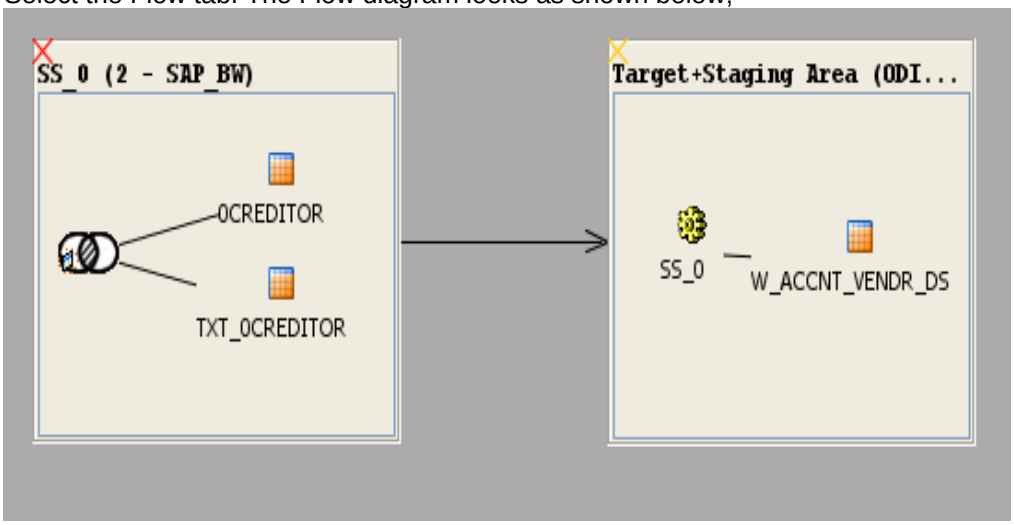


4. Repeat this operation to perform the following simple mappings:
   - **W_ACCNT_VENDR_DS.PCOMPANY = 0CREDITOR.PCOMPANY**
   - **W_ACCNT_VENDR_DS.PHONE = 0CREDITOR.PHONE**
   - **W_ACCNT_VENDR_DS.PLANT = 0CREDITOR.PLANT**
   - **W_ACCNT_VENDR_DS.STREET = 0CREDITOR.STREET**
   - **W_ACCNT_VENDR_DS.MED_DESC = TXT_0CREDITOR.TXTMD**

Define the Interface Flow

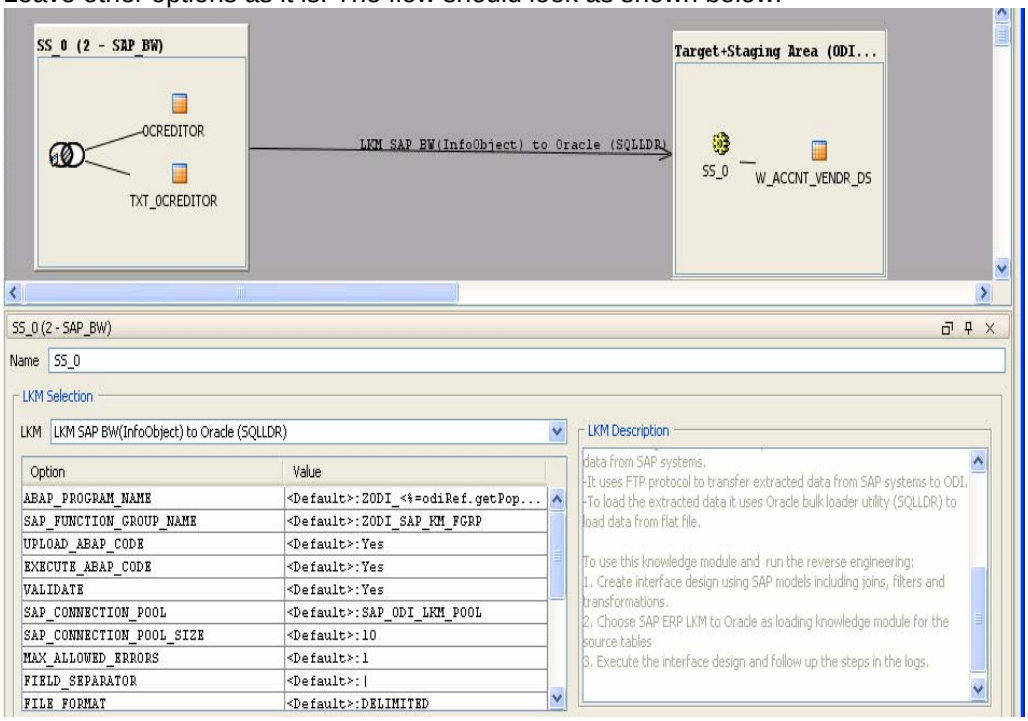
To define the interface flow:

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



2. Select the source set `SS_0 (SAP_BW)` that contains both the Info Object's main table `0CREDITOR` and text table `TXT_0CREDITOR`. This source set represents the source dataset made up of Info Object's main table & text table joined `SAP BW` data targets.
3. In the LKM Selection box, select `LKM SAP BW to Oracle (SQLLDR)`.
4. Select the **Target + Staging Area** box. This represents the target Oracle database, which is also used as the staging area for this interface.
5. In the IKM Selection, select `IKM Oracle Incremental Update`.

6. Set the IKM options as follows:
    a. Select `No` for the *FLOW_CONTROL* option.
    b. Leave other options as it is. The flow should look as shown below:



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

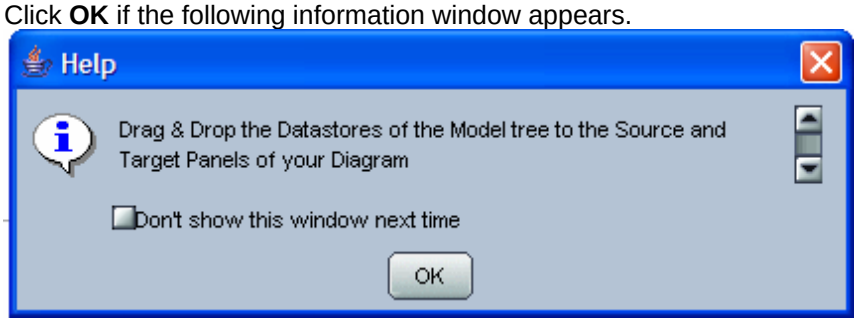
ODS/DSO Extraction

This section describes how to create an integration interface 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.

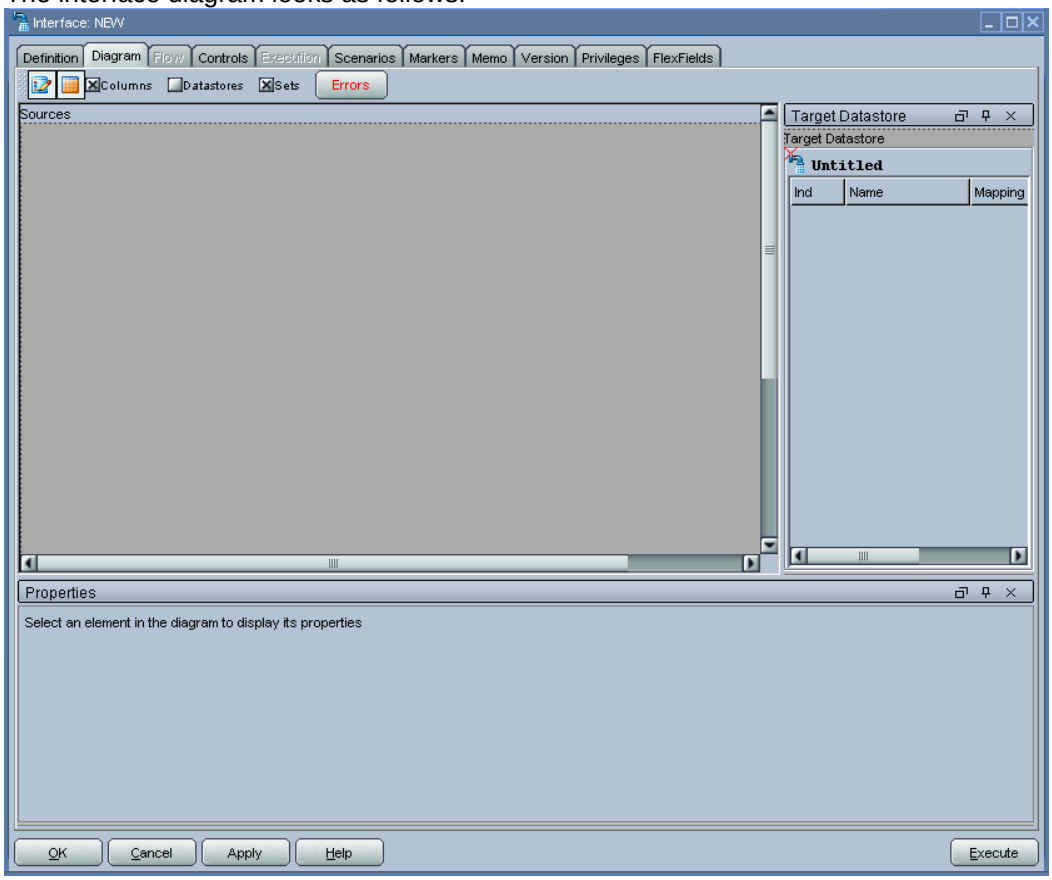
Create the Interface

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 Interfaces node.
5. Right-click and select **Insert Interface**.
6. In the Interface Definition tab, enter the interface name: `ODS_SAP_BW_PurchOrdl.W_PURCH_DS`.
7. Select the Diagram tab.

8. Click **OK** if the following information window appears.



The interface diagram looks as follows:

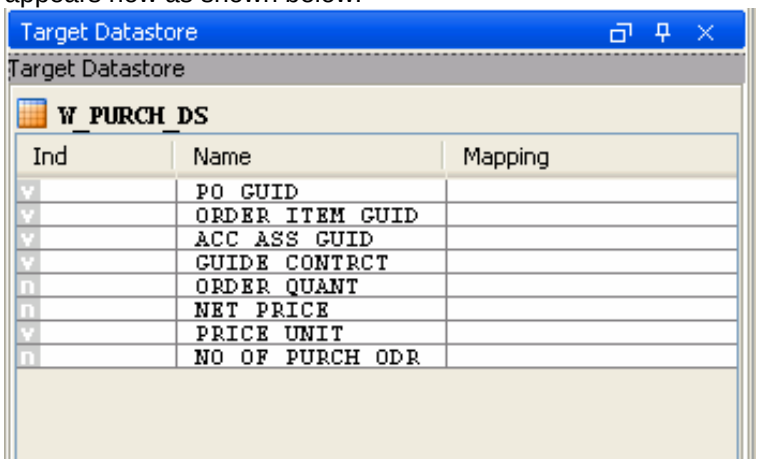


Define the Source and Target Datastores

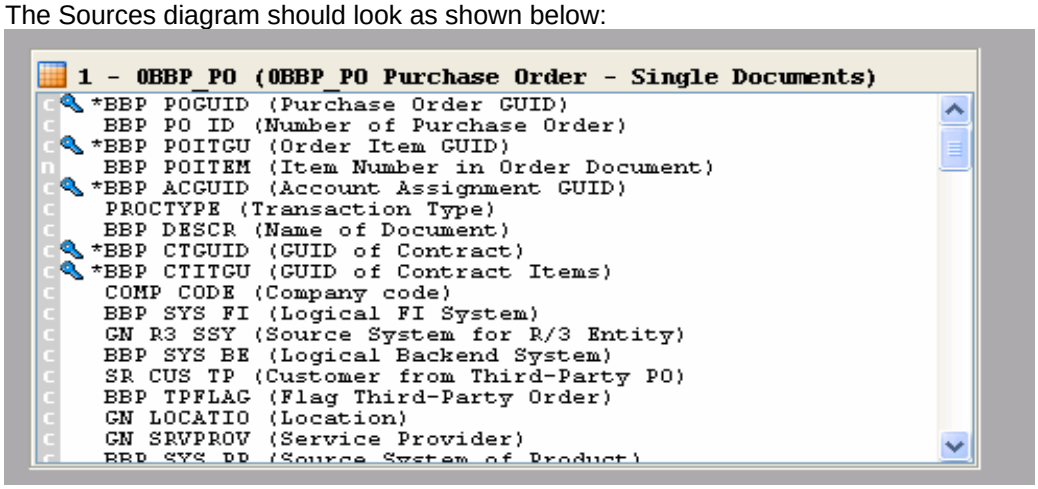
To define the source and target datastores:

1. In the Models tree view, select the `W_PURCH_DS` datastore from the `Oracle_Target` Model.

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



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` diagram panel (left area of the diagram tab).
5. If a popup window prompting “Do you want to perform Automatic Mapping?” appears, click **No**.
6. The Sources diagram should look as shown below:

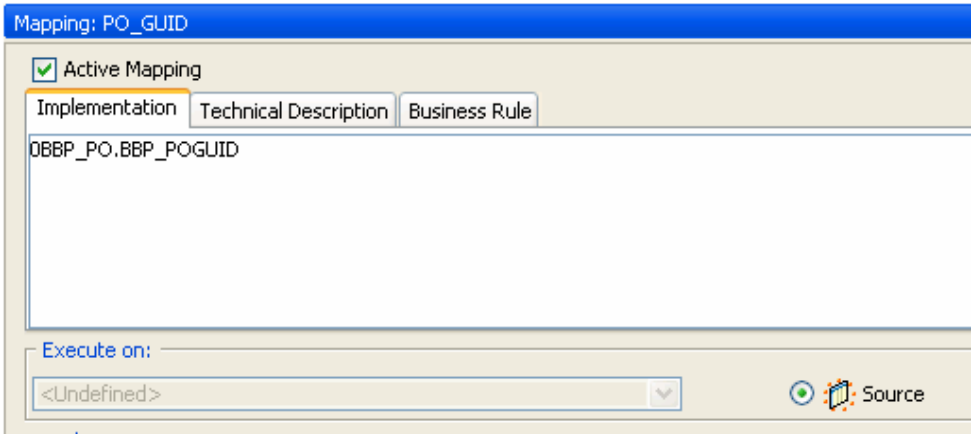


Create the Mappings

To create the mappings of the target datastore:

1. Select the `PO_GUIDE` column from the target datastore.
2. Drag the `BBP_POGUID` column from the `0BBP_PO Purchase Order - Single Documents` source data target into the Implementation field of the Properties panel.

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

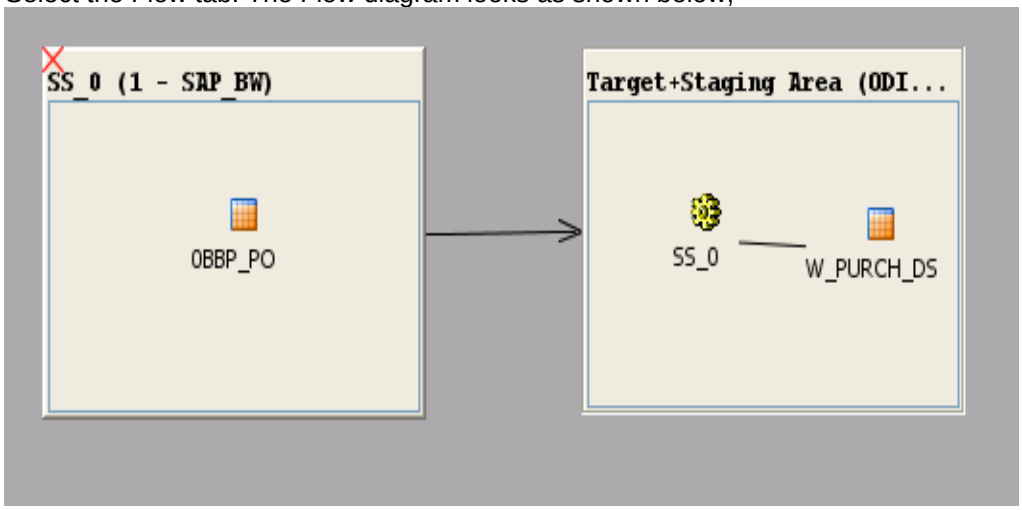


4. Repeat this operation to perform the following simple mappings:
   - **`W_PURCH_DS.ORDER_ITEM_GUID = 0BBP_PO.BBP_POITGU`**
   - **`W_PURCH_DS.ACC_ASS_GUID = 0BBP_PO.BBP_ACGUID`**
   - **`W_PURCH_DS.GUIDE_CONTRCT = 0BBP_PO.BBP_CTGUID`**
   - **`W_PURCH_DS.ORDER_QUANT = 0BBP_PO.BBP_ASPQOU`**
   - **`W_PURCH_DS.NET_PRICE = 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 Interface Flow

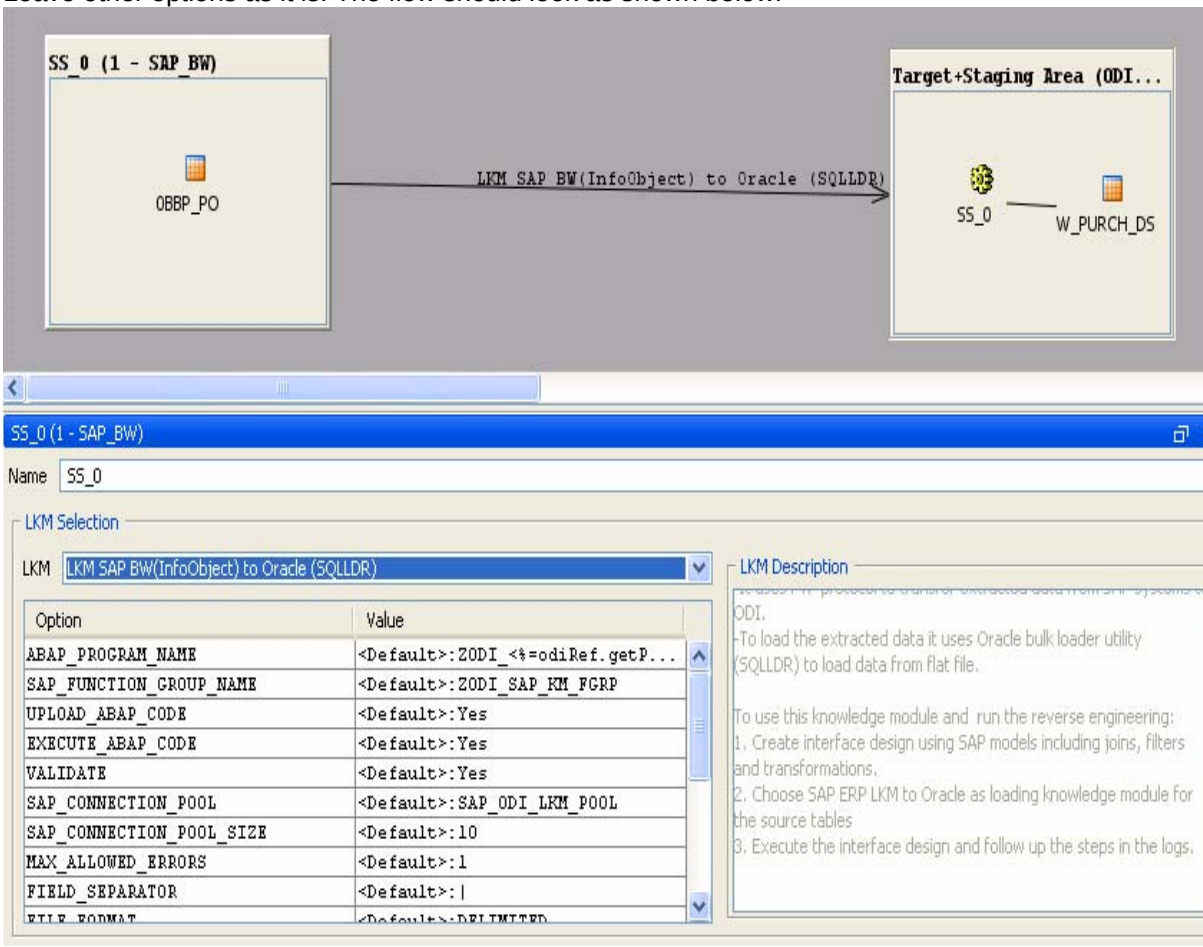
To define the interface flow:

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



2. Select the source set `SS_0 (SAP_BW)` that contains ODS/DSO `0BBP_PO` data targets. This source set represents the source dataset made up of `SAP BW` data target.
3. In the LKM Selection box, select `LKM SAP BW to Oracle (SQLLDR)`.
4. Select the **Target + Staging Area** box. This represents the target Oracle database, which is also used as the staging area for this interface.

5. In the IKM Selection, select `IKM Oracle Incremental Update`.
6. Set the IKM options as follows:
    a. Select `No` for the *FLOW_CONTROL* option.
    b. Leave other options as it is. The flow should look as shown below:



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

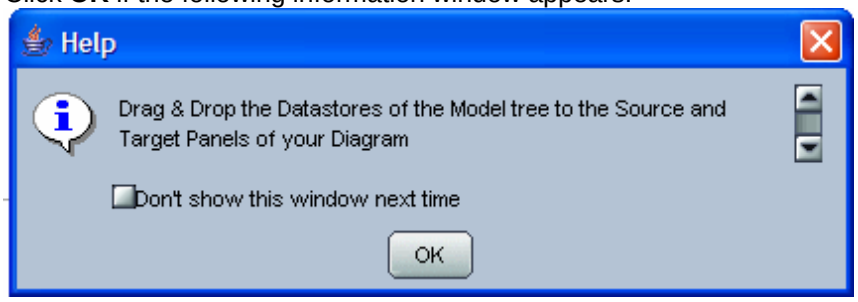
Info Cube & related Info Object Extraction

This section describes how to create an integration interface 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.

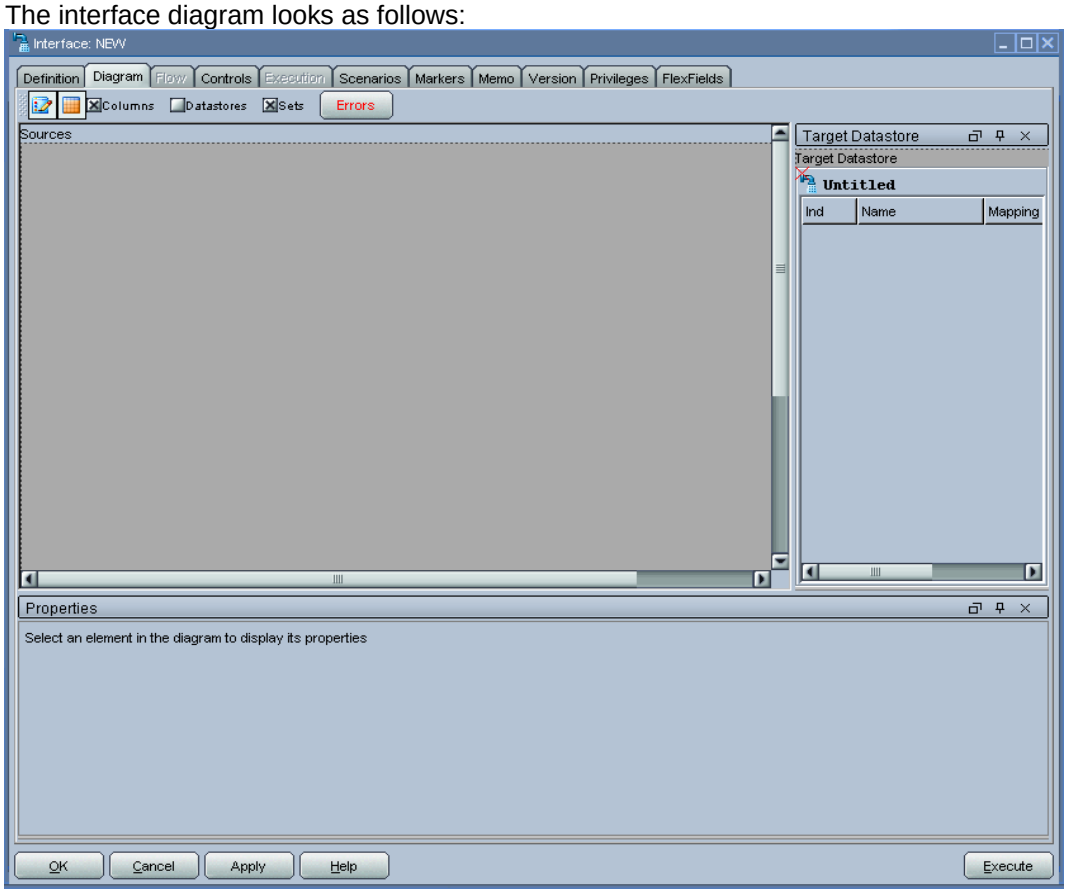
Create the Interface

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 Interfaces node.
5. Right-click and select **Insert Interface**.
6. In the Interface Definition tab, enter the interface name: `InfoCube_InfoObject_join.W_VENDER_CUBE_OBJ_DS`.
7. Select the Diagram tab.
8. Click **OK** if the following information window appears.



The interface diagram looks as follows:

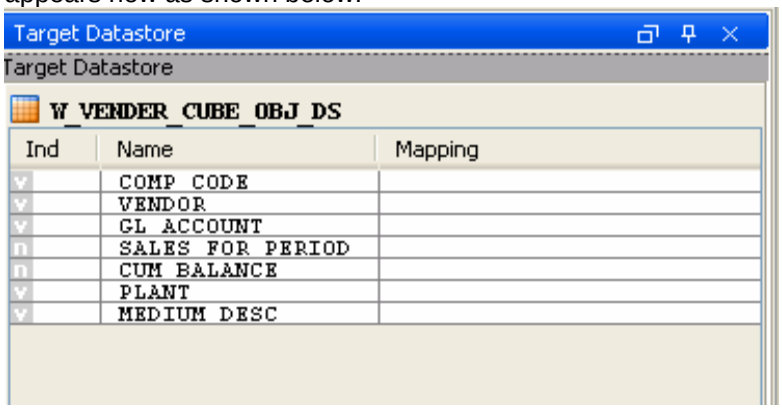


Define the Source and Target Datastores

To define the source and target datastores:

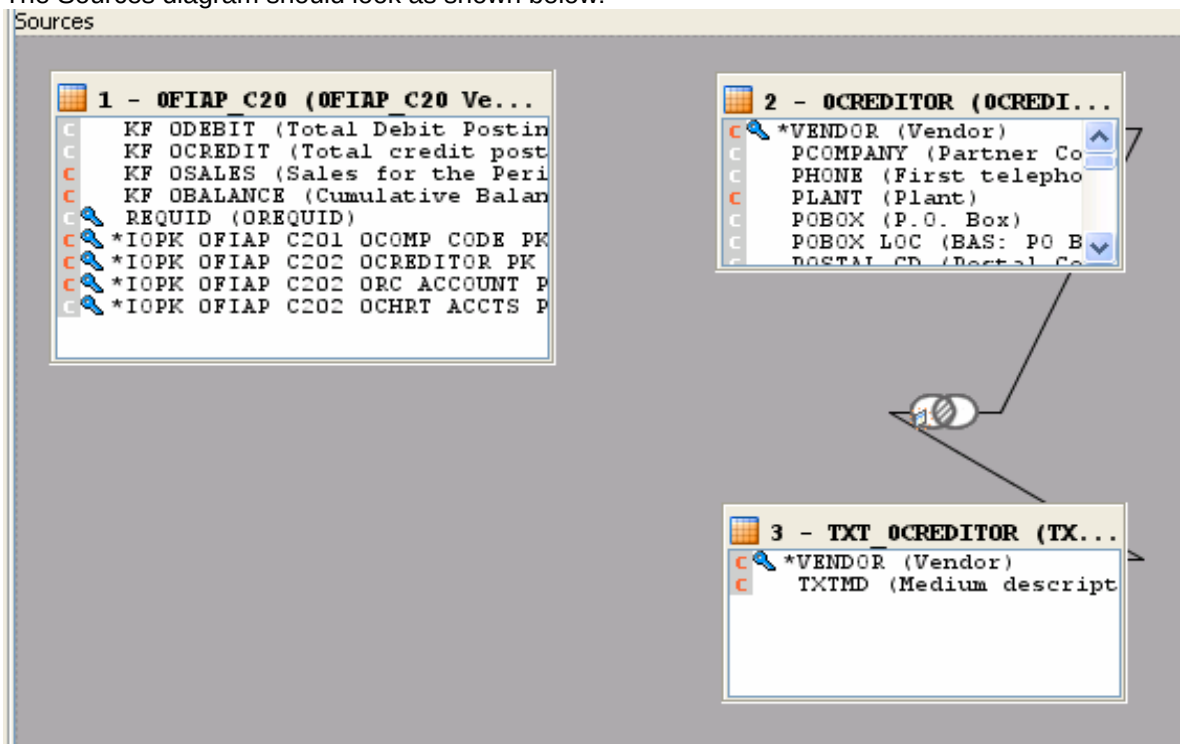
1. In the Models tree view, select the `W_VENDER_CUBE_OBJ_DS` datastore from the `Oracle_Target` Model.

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



3. In the Models tree view, select the Info Cube `0FIAP_C20 Vendor Balances` datastore from the SAP `BW Source` model.
4. Drag this datastore into the `Sources` diagram panel (left area of the diagram tab).
5. If a popup window prompting "Do you want to perform Automatic Mapping?" appears, click **No**.
6. Perform the same operation for the Info Object's main table `0CREDITOR Vendor` and text table `TXT_0CREDITOR Texts: Char. Vendor` datastore from the SAP `BW Source` model.

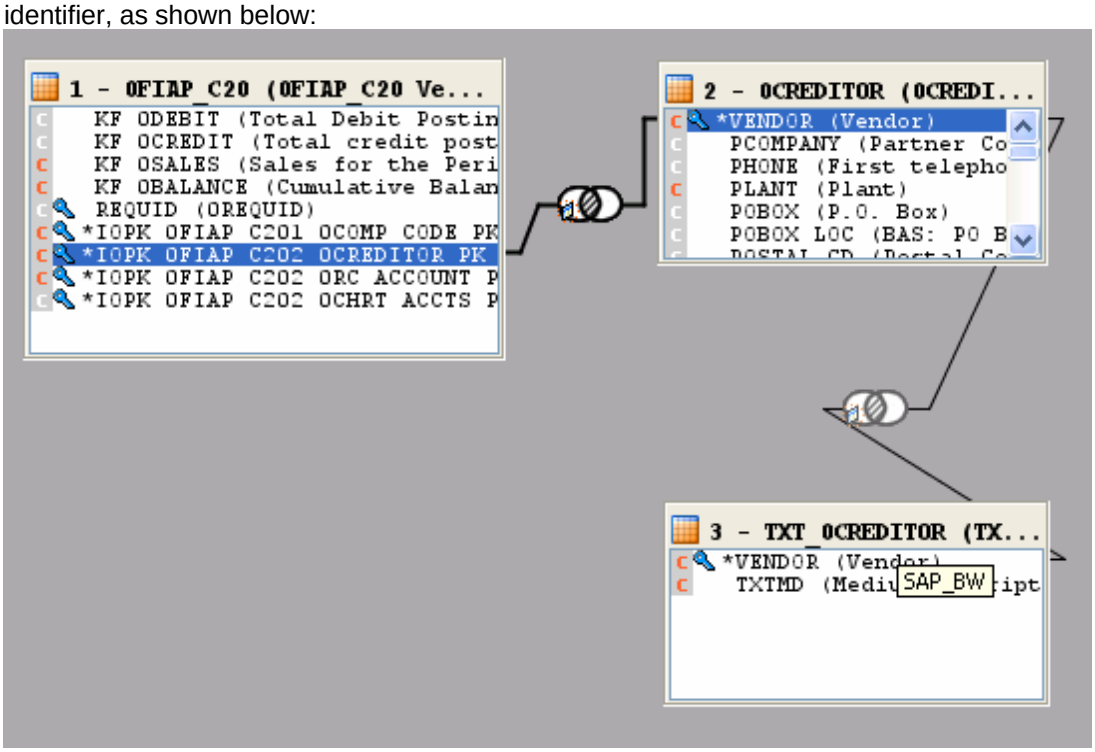
The Sources diagram should look as shown below:



Define Joins between Sources (Info Cube & Object)

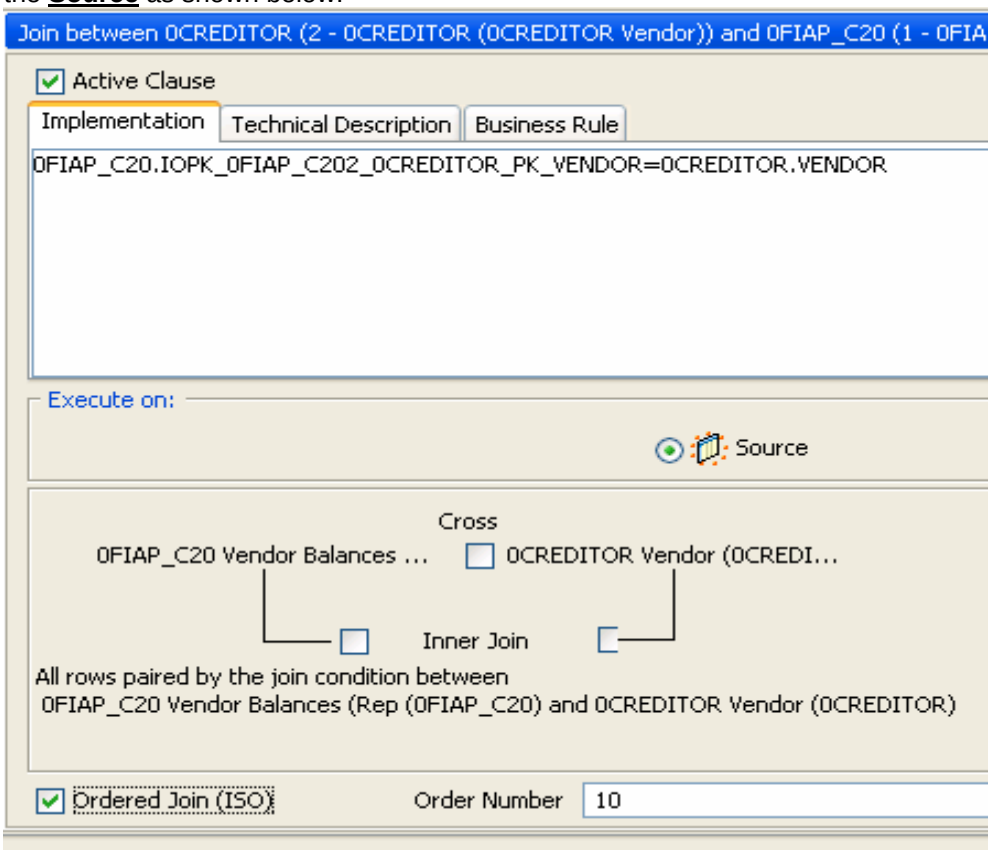
To create joins between the source datastores of an interface:

1. In the Sources diagram drag the `IOPK_0FIAP_C202_0CREDITOR_PK_VENDOR` column from the Info Cube `0FIAP_C20 Vendor Balances` datastore onto the `VENDOR` column 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 `Ordered join` and make sure that this join is executed on the **Source** as shown below:

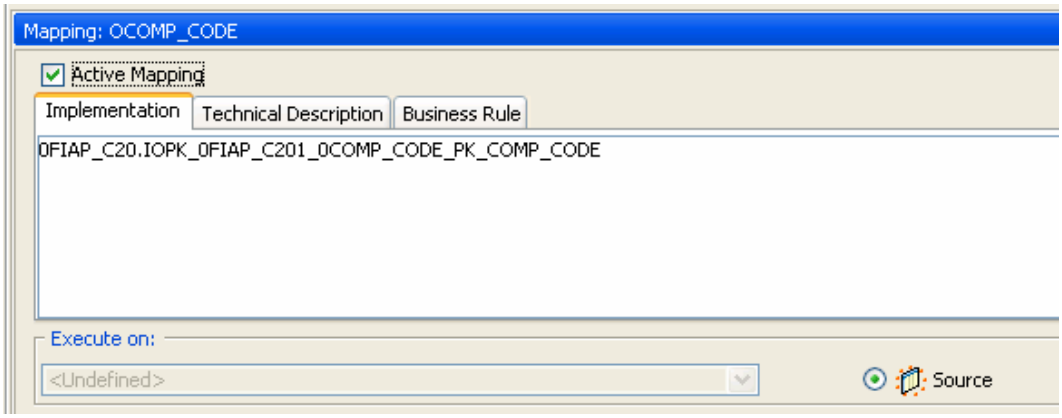


4. Make sure that table number of `0FIAP_C20` is smaller than that of `0CREDITOR`. In *1:n relationships* the parent data target has to be the first in a join.

Create the Mappings

To create the mappings of the target datastore:

1. Select the `COMP_CODE` column from the target datastore.
2. Drag the `IOPK_0FIAP_C201_0COMP_CODE_PK_COMP_CODE` column from the `0FIAP_C20 Vendor Balances` source data target into the Implementation field of the Properties panel.
3. Make sure that the **Execution On** is set to **Source**. The mapping should look as shown below:

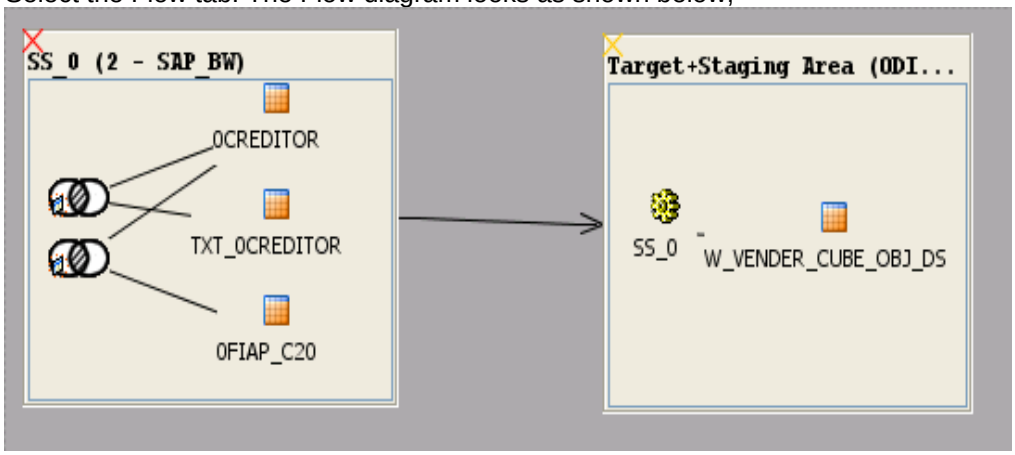


4. Repeat this operation to perform the following simple mappings:
   - **W_VENDER_CUBE_OBJ_DS.VENDOR = 0FIAP_C20.IOPK_0FIAP_C202_0CREDITOR_PK_VENDOR**
   - **W_VENDER_CUBE_OBJ_DS.GL_ACCOUNT = 0FIAP_C20.IOPK_0FIAP_C202_0RC_ACCOUNT_PK_GL_ACCOUNT**
   - **W_VENDER_CUBE_OBJ_DS.SALES_FOR_PERIOD = 0FIAP_C20.KF_0SALES**
   - **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 Interface Flow

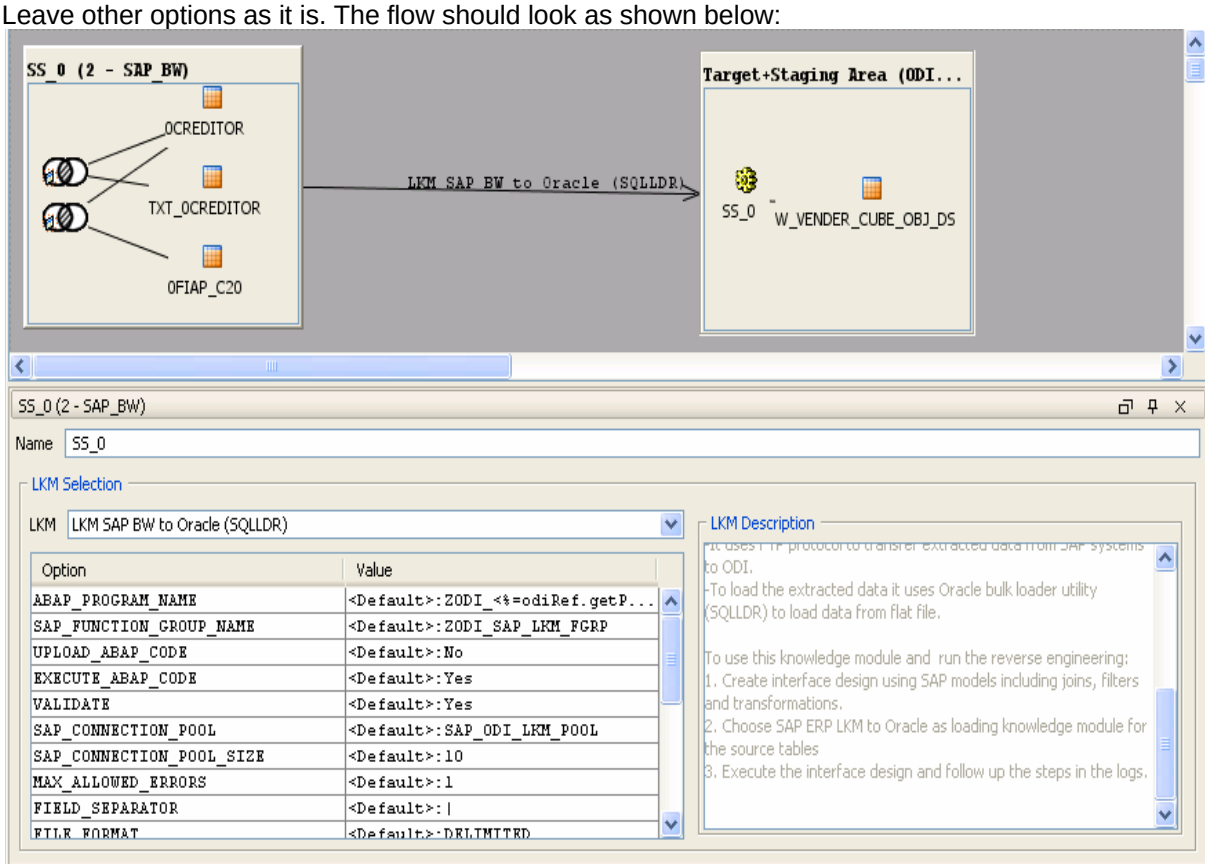
To define the interface flow:

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



2. Select the source set `SS_0 (SAP_BW)` that contains both the `Info Cube` and Info Object tables. This source set represents the source dataset made up of two joined SAP `BW` data targets.
3. In the LKM Selection box, select `LKM SAP BW to Oracle (SQLLDR)`.
4. Select the **Target + Staging Area** box. This represents the target Oracle database, which is also used as the staging area for this interface.
5. In the IKM Selection, select `IKM Oracle Incremental Update`.
6. Set the IKM options as follows:
   a. Select `No` for the *FLOW_CONTROL* option.

b. Leave other options as it is. The flow should look as shown below:



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

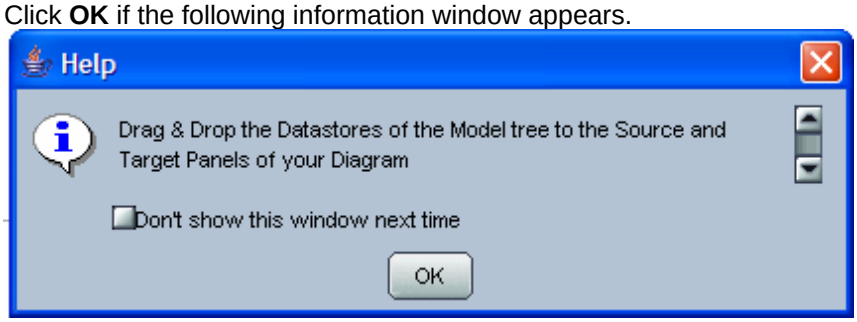
Open Hub Extraction

This section describes how to create an integration interface in ODI in order to load data from a SAP BW source model representing the Open Hub destination `ZOHD_SALE (/BIC/OHZOHD_SALE)` to the `W_OPENHUB_SALES_DS` Oracle target model.

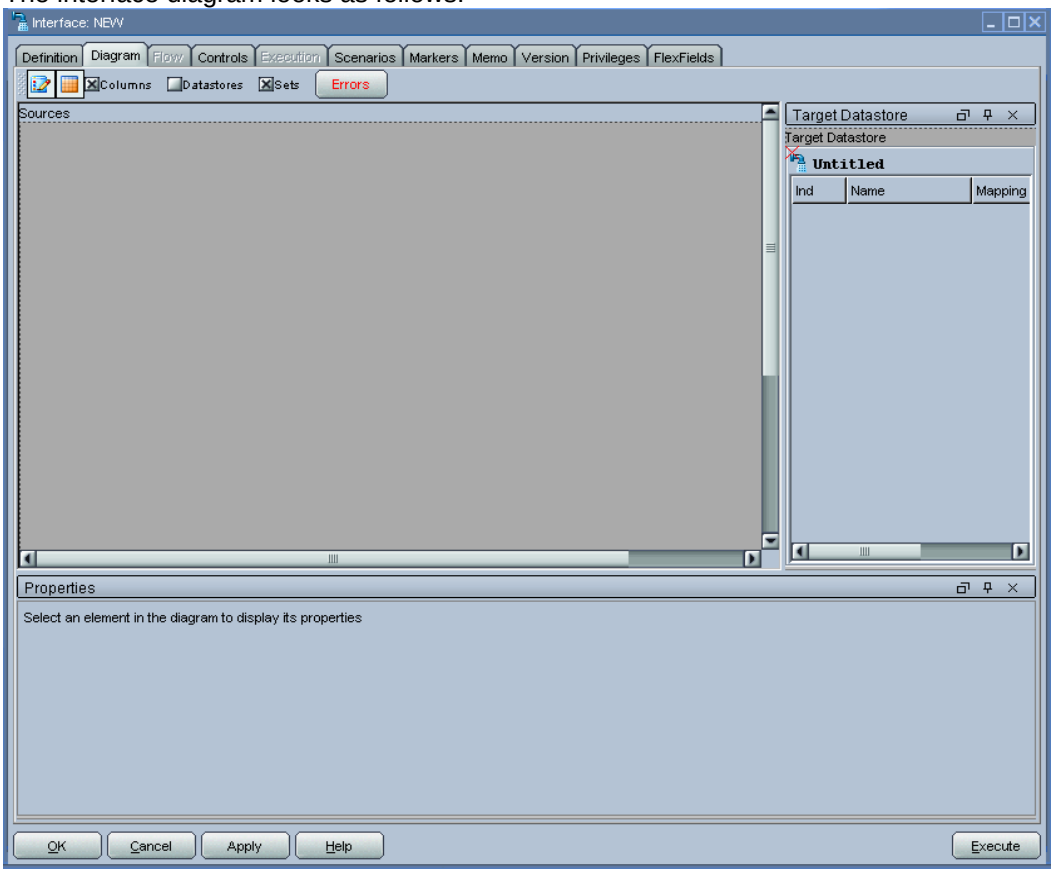
Create the Interface

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 Interfaces node.
5. Right-click and select **Insert Interface**.
6. In the Interface Definition tab, enter the interface name: `OpenHub_Sales.W_OPENHUB_SALES_DS`.
7. Select the Diagram tab.

8. Click **OK** if the following information window appears.



The interface diagram looks as follows:

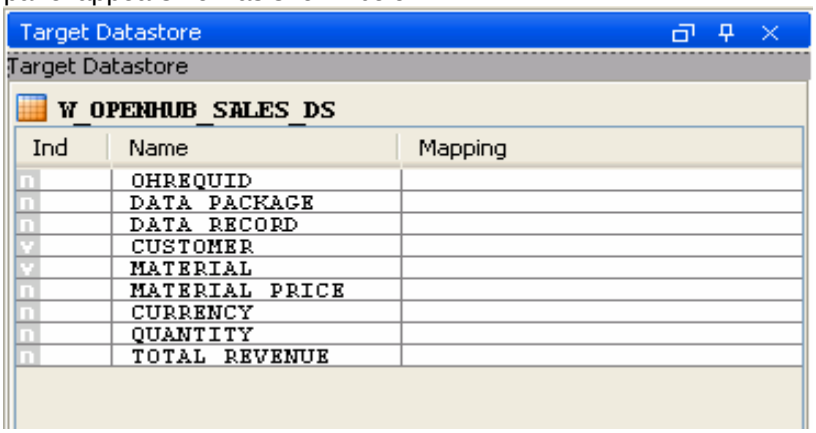


Define the Source and Target Datastores

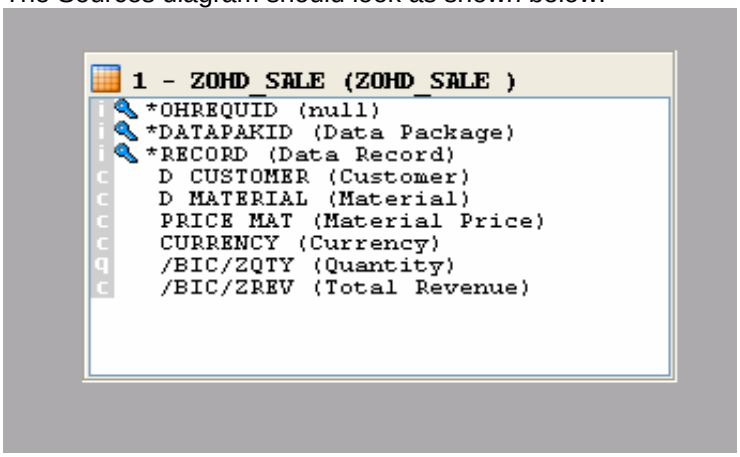
To define the source and target datastores:

1. In the Models tree view, select the `W_OPENHUB_SALES_DS` datastore from the `Oracle_Target` Model.

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



3. In the Models tree view, select the `ZOHD_SALE` datastore from the `SAP BW Source` model.
4. Drag this datastore into the `Sources` diagram panel (left area of the diagram tab).
5. If a popup window prompting “Do you want to perform Automatic Mapping?” appears, click **No**.
6. The Sources diagram should look as shown below:

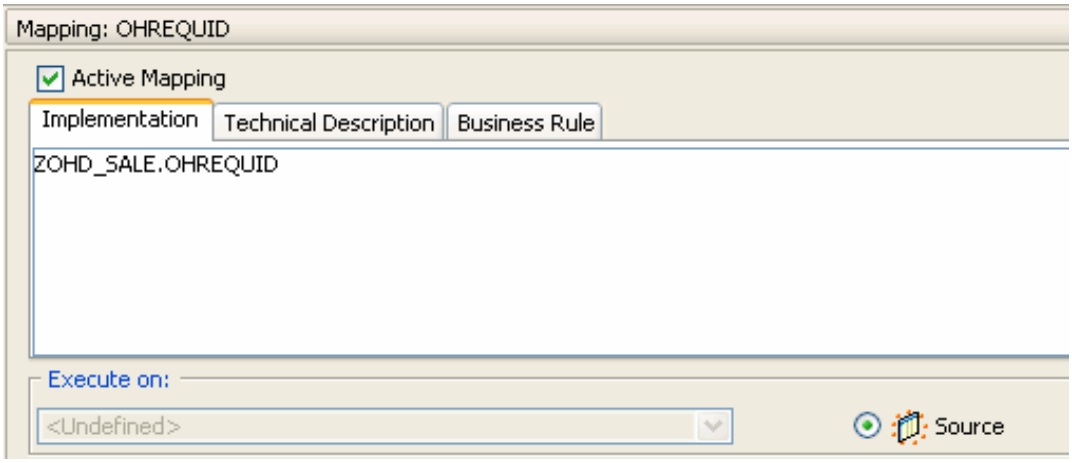


Create the Mappings

To create the mappings of the target datastore:

1. Select the `OHREQUID` column from the target datastore.
2. Drag the `OHREQUID` column from the `ZOHD_SALE` source data target into the Implementation field of the Properties panel.

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

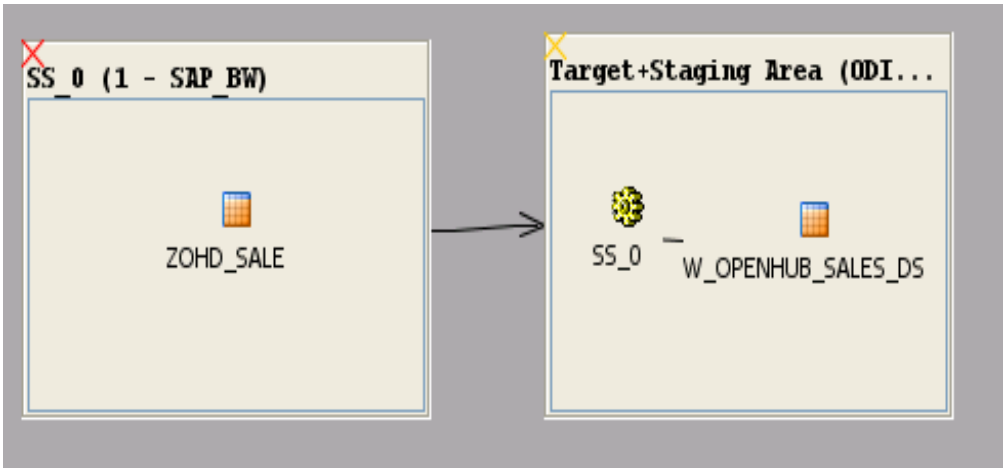


4. Repeat this operation to perform the following simple mappings:
   - **W_OPENHUB_SALES_DS.DATA_PACKAGE = ZOHD_SALE.DATAPAKID**
   - **W_OPENHUB_SALES_DS.DATA_RECORD = ZOHD_SALE.RECORD**
   - **W_OPENHUB_SALES_DS.CUSTOMER = ZOHD_SALE.D_CUSTOMER**
   - **W_OPENHUB_SALES_DS.MATERIAL = ZOHD_SALE.D_MATERIAL**
   - **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 Interface Flow

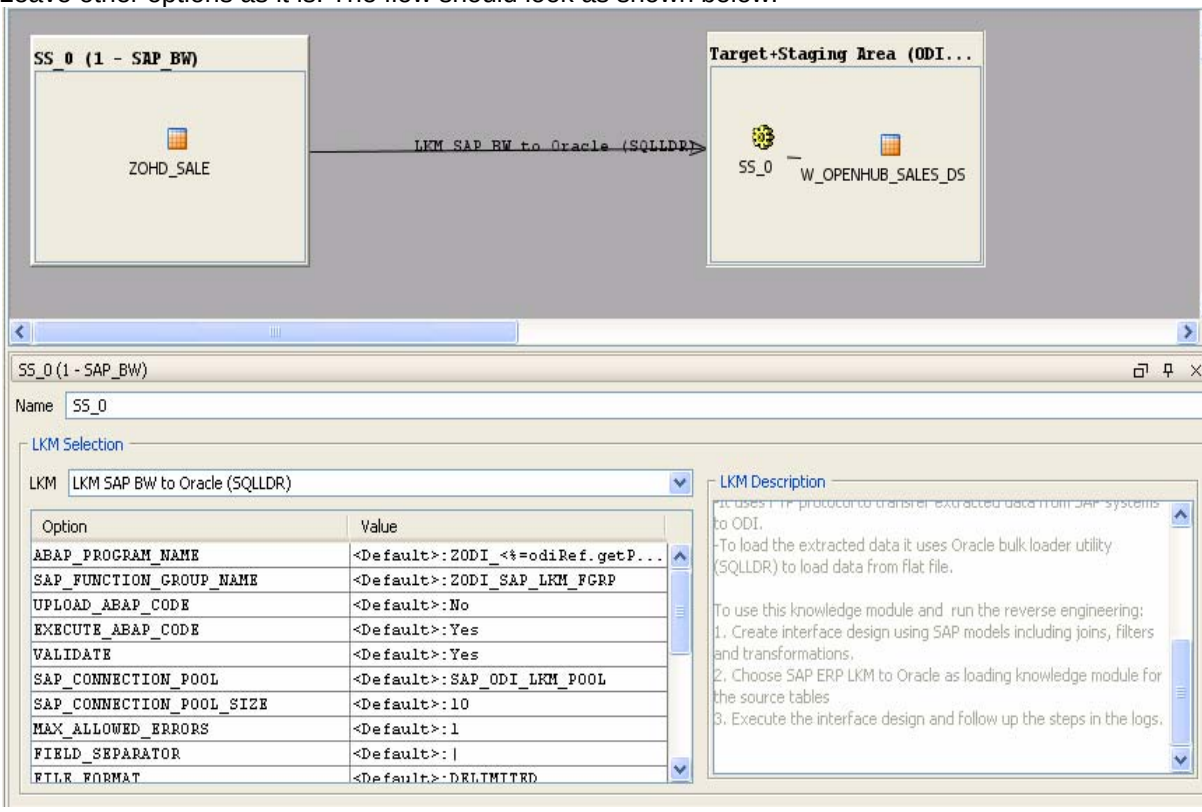
To define the interface flow:

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



2. Select the source set `SS_0 (SAP_BW)` that contains Open Hub Destination Table `ZOHD_SALE`. This source set represents the source dataset made up of SAP BW Open Hub table.

3. In the LKM Selection box, select `LKM SAP BW to Oracle (SQLLDR)`.
4. Select the **Target + Staging Area** box. This represents the target Oracle database, which is also used as the staging area for this interface.
5. In the IKM Selection, select `IKM Oracle Incremental Update`.
6. Set the IKM options as follows:
    a. Select `No` for the *FLOW_CONTROL* option.
    b. Leave other options as it is. The flow should look as shown below:



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

Running the Integration Interface

Running the Interface

To run the integration interface:

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

What happens when you run the interface?

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

1. It performs the extraction of the data on the SAP 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.

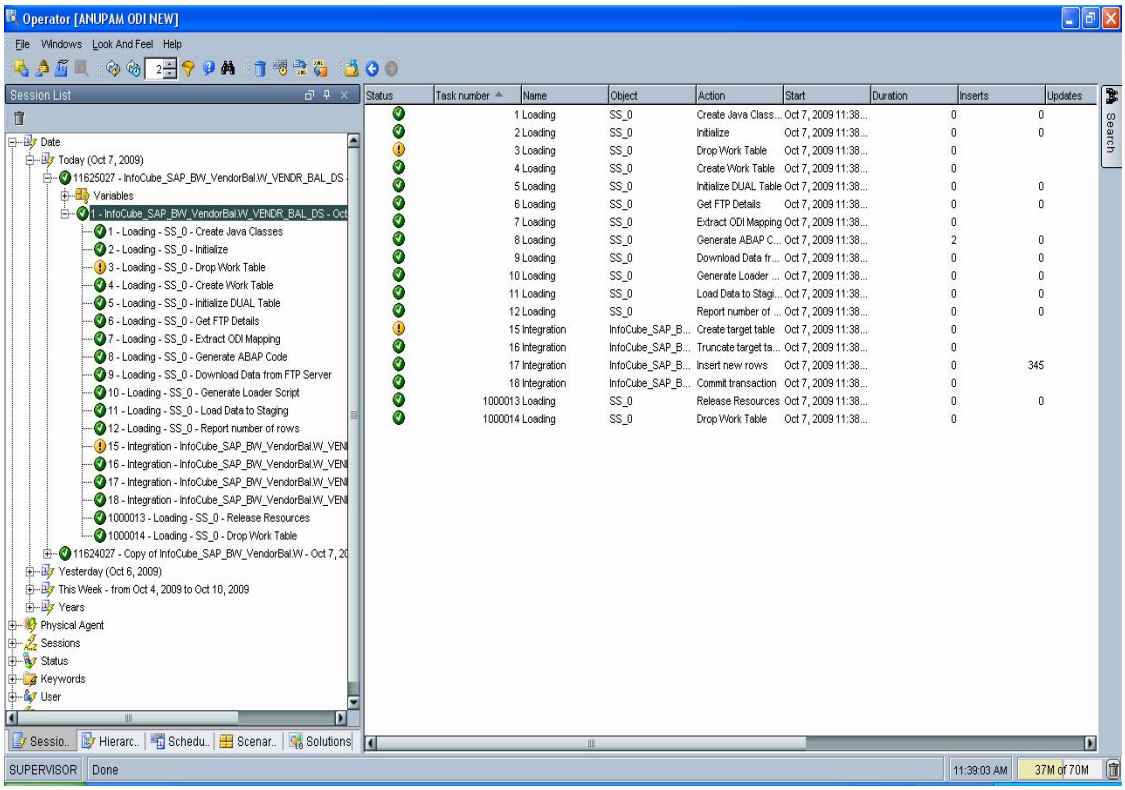
Finally, when the agent accesses the extraction file, it uses *SQL*Loader* to load this file into the Oracle staging area. The rest of the integration process takes place within the Oracle engine.

Review the Interface Execution

To review the interface execution:

1. Connect to Operator.

2. In the Operator, select the Session List tree view.
3. Expand the All Executions node in this tree view.
4. The latest session is the first entry of this list and appears as shown below.



5. Double click the `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 interfaces:
    - InfoObject_SAP_BW_VendorAcc.W_ACCNT_VENDR_DS
    - ODS_SAP_BW_PurchOrdl.W_PURCH_DS
    - InfoCube_InfoObject_join.W_VENDER_CUBE_OBJ_DS
    - OpenHub_Sales.W_OPENHUB_SALES_DS
    - Delta Extraction: Execute the interface after full load with FIRST_REQ_ID and LAST_REQ_ID value for delta load. The new data will loaded in the above oracle table as per provided request ID.

Creating a Package for Delta Extraction

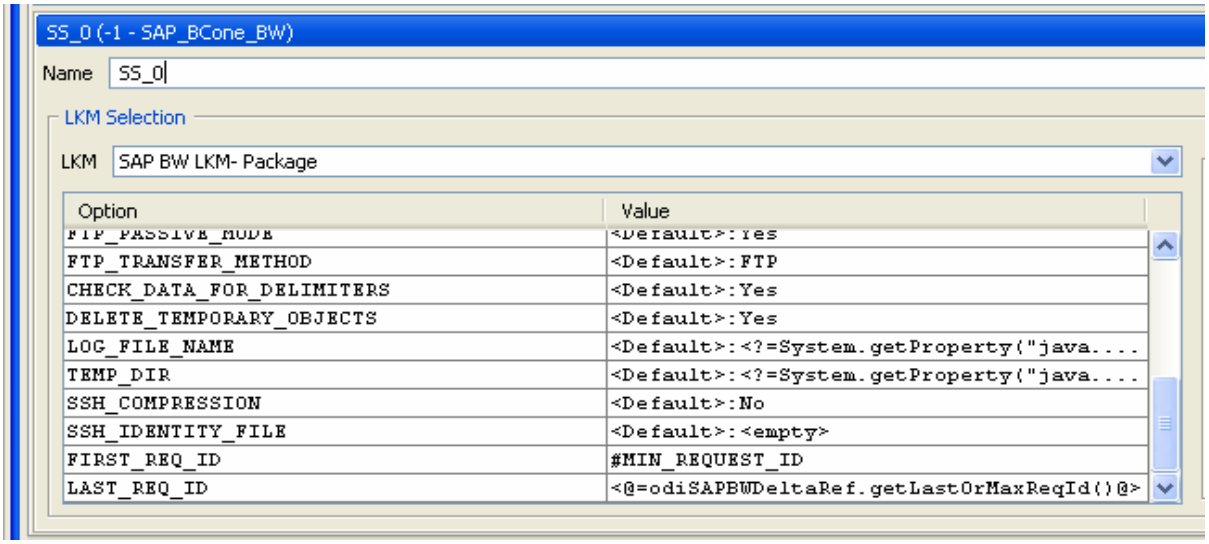
So far we have seen examples of extracting data from BW data targets. The instructions in this chapter explain how the interfaces introduced in the previous chapters can be turned into *delta extraction interfaces*.

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

Modify the Interface Flow

The first step is to modify the interface:

1. Select the desired Interfaces.
2. Double click the interface.
3. Select the **Flow** tab.
4. Click the ODI source set.
5. Set KM option values for FIRST_REQ_ID and LAST_REQ_ID KM as shown below.

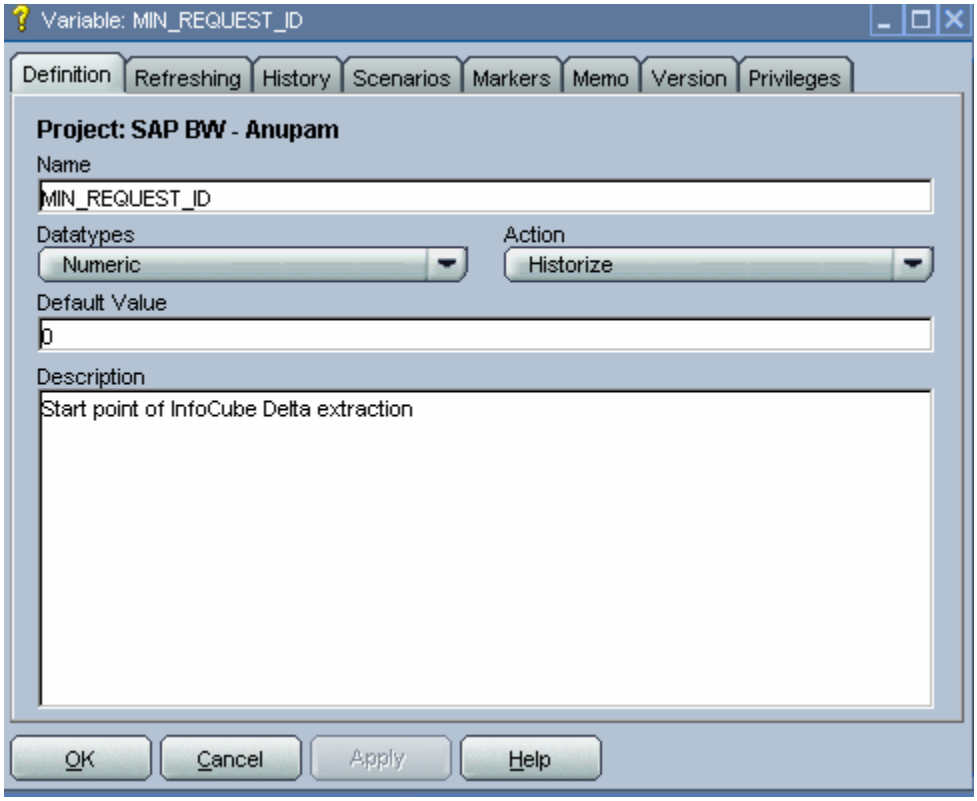


6. Click **OK** to save and close your interface.

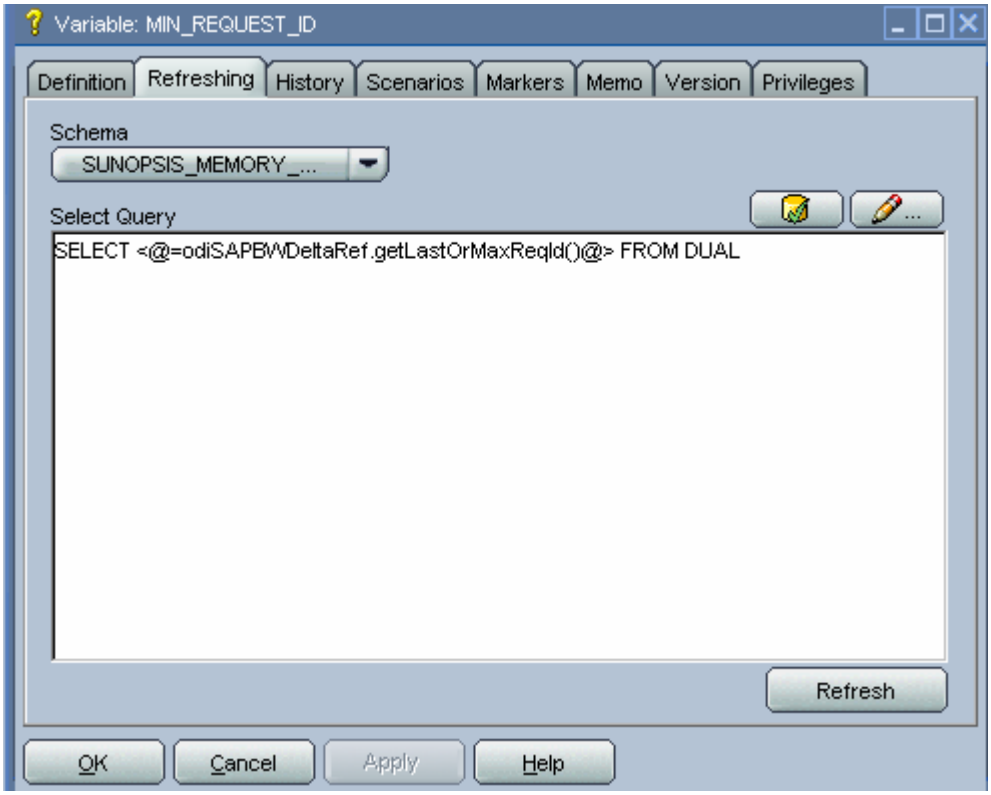
Creating the MIN_REQUEST_ID Variable

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 `Historize` or `Last-value` Action as shown below.



2. 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 Interfaces are arranged into a Package:

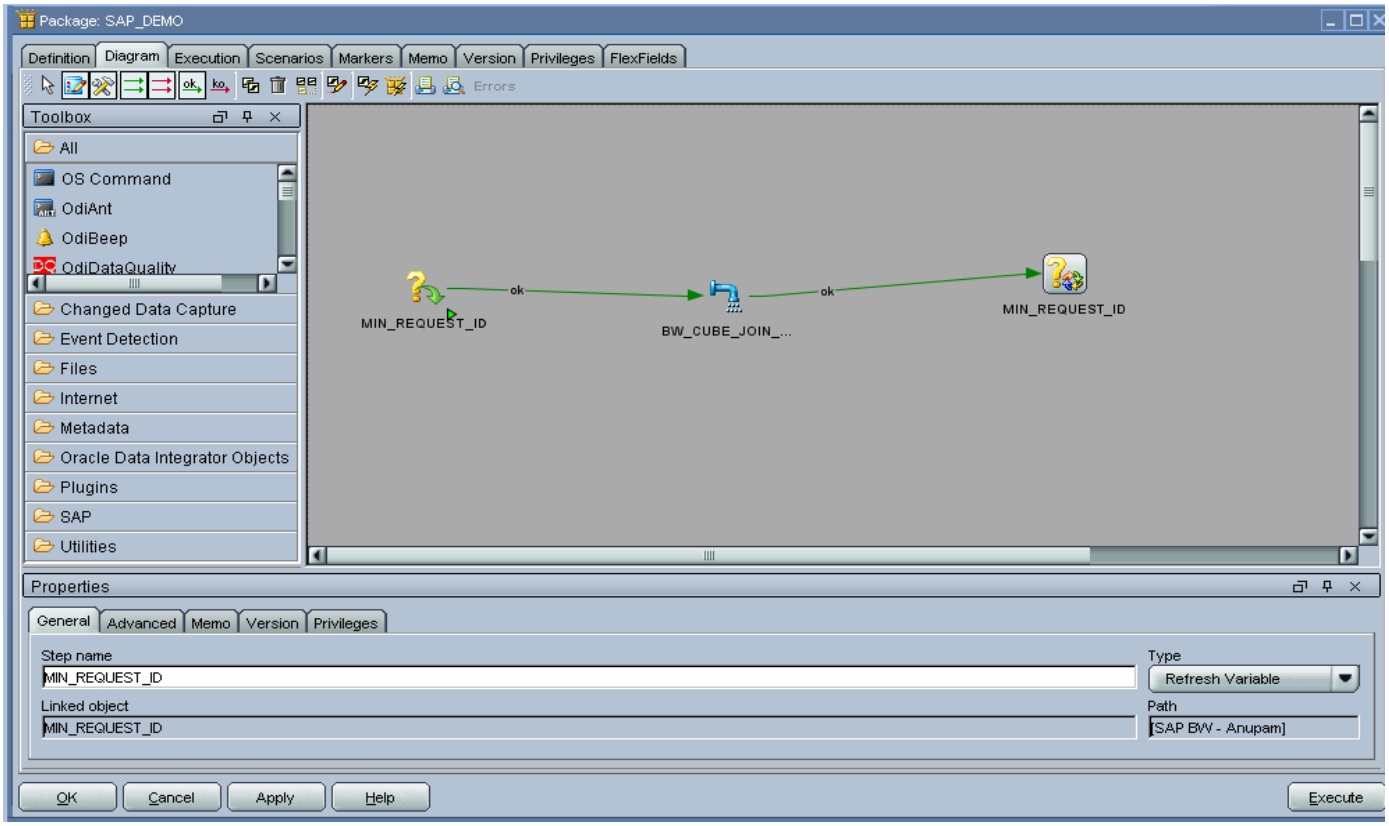
1. Open the `SAP Demo` Project.
2. Create a new Package named `SAP Demo`.
3. Insert a Declare Variable step to this Package:

   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 Interface 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. Definine the step sequence.
   1. Define the Refresh Variable step as the **First Step**.
   2. Link it to the Flow step that is the next step upon success.
   3. Link the 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 integration Package:

1. In the Projects tree view, expand the SAP Demo Project.
2. Select the SAP Demo Package.
3. Right-click and select **Execute.**
4. In the Execution window, click **OK**.
5. 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 (interface).

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

Conclusion

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 interfaces 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 *Oracle Data Integrator Knowledge Modules Reference Guide* for more information on the SAP BW KM features and options.

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

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

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

Going further with Oracle Data Integrator

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

Appendix A - SAP ABAP BW Required Privileges

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.

Later for RKM execution and LKM execution in production a SAP RFC user is sufficient.

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

Authorizations Required for RKM SAP BW Setup

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

Object	Field Name	Value
S_ADMI_FCD	S_ADMI_FCD	*
S_ADMI_FCD	S_ADMI_FCD	NADM
S_ADMI_FCD	S_ADMI_FCD	PADM
S_CTS_ADMI	CTS_ADMFCT	EPS1
S_CTS_ADMI	CTS_ADMFCT	EPS2
S_CTS_ADMI	CTS_ADMFCT	PROJ
S_C_FUNCT	ACTVT	16
S_C_FUNCT	CFUNCNAME	*
S_C_FUNCT	PROGRAM	*
S_CTS_ADMI	CTS_ADMFCT	TABL
S_TCODE	TCD	SU53
S_TCODE	TCD	SE38
S_TCODE	TCD	SE80
S_DATASET	ACTVT	6
S_DATASET	ACTVT	33
S_DATASET	ACTVT	34
S_DATASET	FILENAME	*
S_TCODE	TCD	SE91

S_TCODE	TCD	ST22
S_DATASET	PROGRAM	*
S_RFC	RFC_TYPE	*
S_RFC	RFC_NAME	*
S_RFC	ACTVT	16
S_DEVELOP	ACTVT	*
S_DEVELOP	DEVCLASS	*
S_DEVELOP	OBJNAME	*
S_DEVELOP	OBJTYPE	*
S_DEVELOP	P_GROUP	*
S_DOKU_AUT	DOKU_ACT	RAW_VERS
S_DOKU_AUT	DOKU_DEVCL	TEST
S_DOKU_AUT	DOKU_MODE	MAINTAIN
S_RFC_ADM	ACTVT	3
S_RFC_ADM	ICF_VALUE	*
S_RFC_ADM	RFCDEST	*
S_RFC_ADM	RFCTYPE	*
S_RZL_ADM	ACTVT	*
S_TABU_DIS	ACTVT	3
S_TABU_DIS	DICBERCLS	*
S_TCODE	TCD	▪ AL11
S_TCODE	TCD	SE10
S_TCODE	TCD	SE11
S_TCODE	TCD	SE16
S_TCODE	TCD	SE37
S_TCODE	TCD	SM58
S_TCODE	TCD	SM59
S_TRANSPRT	ACTVT	1
S_TRANSPRT	ACTVT	2
S_TRANSPRT	ACTVT	3
S_TRANSPRT	ACTVT	5

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

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

Authorizations Required for RKM Execution

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

Object	**Field Name**	**Value**
S_ADMI_FCD	S_ADMI_FCD	*

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

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

Authorizations Required for LKM Execution

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

Object	**Field Name**	**Value**
S_ADMI_FCD	S_ADMI_FCD	*
S_ADMI_FCD	S_ADMI_FCD	NADM
S_ADMI_FCD	S_ADMI_FCD	PADM
S_CTS_ADMI	CTS_ADMFCT	EPS1
S_CTS_ADMI	CTS_ADMFCT	EPS2
S_CTS_ADMI	CTS_ADMFCT	PROJ
S_C_FUNCT	ACTVT	16

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

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

Authorizations Required for LKM Execution for Production

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

Object	Field Name	Value
S_ADMI_FCD	S_ADMI_FCD	*

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

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

Appendix B - SAP Stand-Alone Connection Test

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

To use a standalone java utility to test the connection:

1. Open a command window.
2. Change to `oracledi/drivers` directory.
3. Make sure that JAVA_HOME points to a JVM (at least 1.5).
4. Make sure that you have installed SAP Java Connector and that the `sapjar3.jar` and the sapjco3 library are in the `oracledi/drivers` directory.
5. Launch the utility using the following command:
   ```
   java –cp sapjarco3.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 fail
Connection parameters: TYPE=A DEST=ODI_SAP_CON_POOL ASHOST=<SAP Application Server> SYSNR=00 PCS

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
```

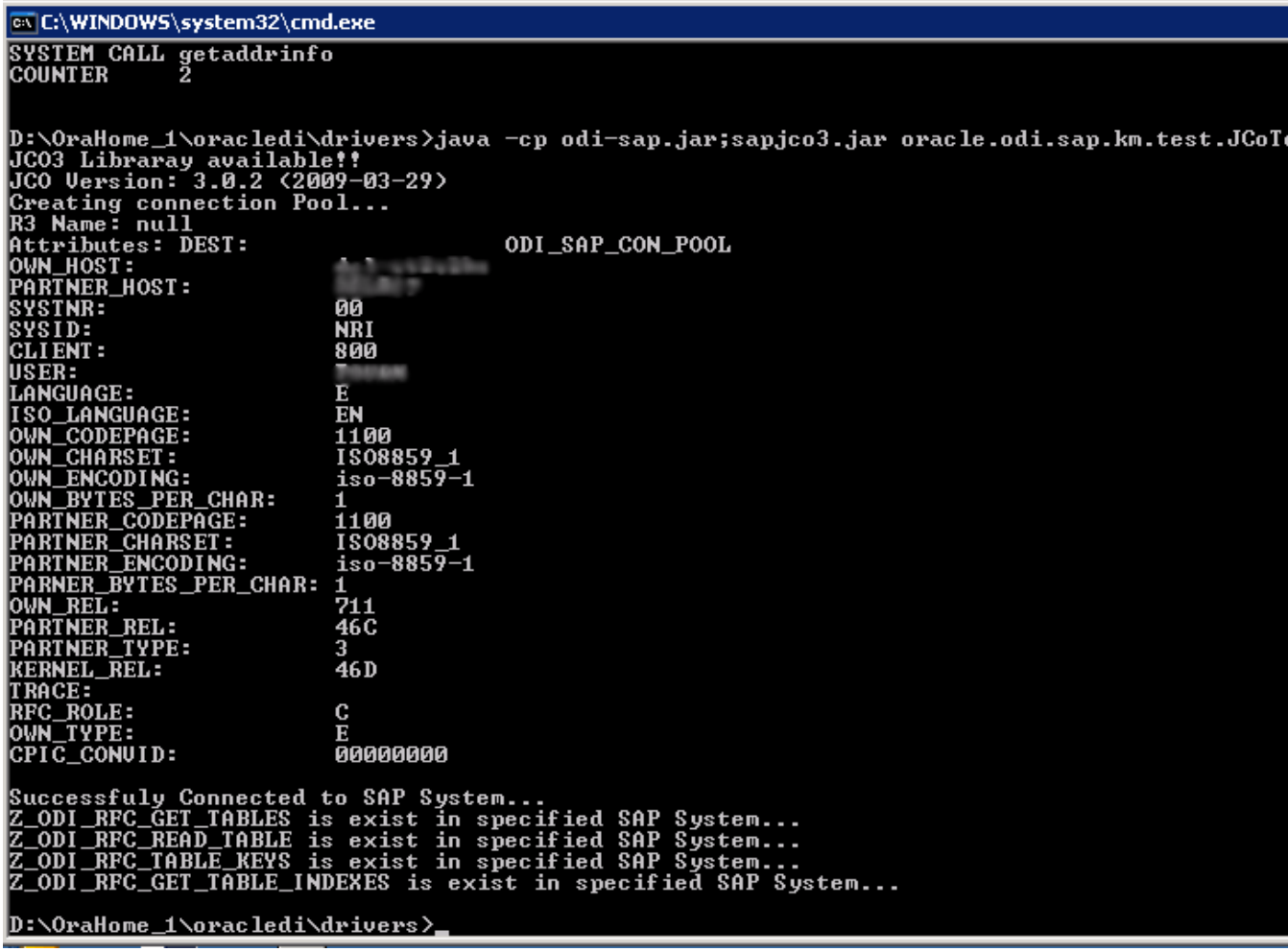
6. Use a text editor to open the `ODI_SAP_CON_POOL.jcoDestination` file in the `oracledi/drivers` directory. The file should look like this:

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

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

8. Launch the utility again using the following command:
   `java –cp sapjarco3.jar;odi-sap.jar oracle.odi.sap.km.test.JCoTest`
   This should result in an output similar to this:



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

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

Appendix C – Uninstalling ODI SAP Components

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

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

To upgrad 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 transaction `SE80`.
4. Select the package or development class option from the drop down box.

5. Type `ZODIBW_PCKG` in the box provided below the drop down box .
6. Right-click the `ZODIBW_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. The system will not allow deleting the development class since `TADIR` entries are there.
9. Go to transaction `SE38`.
10. Create a Z Program.
11. Write the code: `DELETE FROM TADIR WHERE DEVCLASS = ' ZODIBW_PCKG '.`
12. Save, activate and execute the program.
13. Now Go to transaction SE80 and delete the development class ZODIBW_PCKG