

Oracle® Fusion Middleware

Application Adapters Guide for Oracle Data Integrator



12 c12.2.1.4.0

E95628-01

September 2019

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

ORACLE®

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

Primary Author: Oracle Corporation

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

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

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

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

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

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

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	x
Documentation Accessibility	x
Related Documents	x
Conventions	xi

1 Introduction

Terminology	1-1
Using This Guide	1-1

2 Oracle E-Business Suite

Introduction	2-1
Concepts	2-1
Knowledge Modules	2-2
Installation and Configuration	2-2
System Requirements and Certifications	2-2
Technology Specific Requirements	2-3
Connectivity Requirements	2-3
Setting Up the Topology	2-3
Create an Oracle Data Server	2-3
Create an Oracle Physical Schema	2-3
Setting up an Integration Project	2-4
Creating an Oracle Model and Reverse-Engineering E-Business Suite Tables	2-4
Create an Oracle Model	2-4
Reverse-Engineer E-Business Suite Tables	2-5
Designing a Mapping	2-7
Loading Data from E-Business Suite	2-7
Integrating Data in E-Business Suite through the Open Interface	2-7
Managing Group IDs	2-7
Executing an Open Interface Program	2-9

3 JD Edwards EnterpriseOne

Introduction	3-1
Concepts	3-1
Knowledge Modules	3-1
Installation and Configuration	3-2
System Requirements and Certifications	3-2
Technology Specific Requirements	3-2
Connectivity Requirements	3-4
Setting Up the Topology	3-5
Create a Data Server	3-5
Create a Physical Schema	3-5
Setting up an Integration Project	3-5
Creating and Reverse-Engineering a Model	3-6
Create a Model	3-6
Reverse-Engineer JDE Tables	3-6
Designing a Mapping	3-7
Loading Data from JDE EnterpriseOne	3-8
Integrating Data in JDE EnterpriseOne	3-8

4 JD Edwards World

Introduction	4-1
Concepts	4-1
Knowledge Modules	4-1
Installation and Configuration	4-2
System Requirements and Certifications	4-2
Technology Specific Requirements	4-2
Connectivity Requirements	4-3
Setting Up the Topology	4-4
Create a Data Server	4-4
Create a Physical Schema	4-5
Setting up an Integration Project	4-5
Creating and Reverse-Engineering a Model	4-5
Create a Model	4-5
Reverse-Engineer JDE Tables	4-6
Designing a Mapping	4-7
Loading Data from JDE	4-7
Integrating Data in JDE	4-7

5 Oracle PeopleSoft

Introduction	5-1
Concepts	5-1
Knowledge Modules	5-1
Installation and Configuration	5-2
System Requirements and Certifications	5-2
Technology Specific Requirements	5-2
Connectivity Requirements	5-2
Setting up the Topology	5-2
Create a Data Server	5-3
Create a Physical Schema	5-3
Setting up the Project	5-3
Creating and Reverse-Engineering a Model	5-3
Create a Model	5-4
Reverse-Engineer PeopleSoft Tables	5-4
Designing a Mapping	5-4
Loading Data from PeopleSoft	5-5

6 Oracle Siebel CRM

Introduction	6-1
Concepts	6-1
Overview of Extracting Data from Siebel	6-1
Knowledge Modules	6-2
Installation and Configuration	6-3
System Requirements and Certifications	6-3
Technology Specific Requirements	6-4
Connectivity Requirements	6-4
Setting up the Topology	6-4
Create a Data Server	6-4
Create a Physical Schema	6-4
Setting up the Project	6-5
Creating and Reverse-Engineering a Model	6-5
Create a Model	6-5
Reverse-Engineer Siebel CRM Tables	6-5
Designing a Mapping	6-6
Loading Data from Siebel	6-6
Loading Data from Siebel through EIM tables	6-7
Integrating Data in Siebel through EIM tables	6-7

7 SAP ABAP BW

Introduction	7-1
Concepts	7-1
Knowledge Modules	7-2
Overview of the SAP BW Integration Process	7-2
Reverse-Engineering Process	7-3
Integration Process	7-3
Installation and Configuration	7-3
System Requirements and Certifications	7-4
Technology Specific Requirements	7-4
Connectivity Requirements	7-5
Installing and Configuring JCo	7-5
Installing ODI SAP Components into SAP System	7-6
Requesting FTP Server Access	7-6
Validating the SAP Environment Setup	7-7
Gathering SAP Connection Information	7-7
Gathering FTP Connection Information	7-9
Gathering Shared Directory Information	7-9
Adding the Open Tool	7-9
Defining the Topology	7-10
Create the File Data Server	7-10
Create a File Data Server	7-10
Create the File Schema	7-11
Create the SAP ABAP Data Server	7-12
Create the SAP ABAP Data Server	7-12
Create the SAP ABAP Schema	7-14
Setting up the Project	7-14
Creating and Reverse-Engineering a Model	7-14
Creating a SAP BW Model	7-15
Reverse-Engineering a SAP BW Model	7-15
Reverse-Engineering a PSA Table in SAP BW Model	7-15
Designing a Mapping	7-16
Considerations for SAP BW Integration	7-17
File Transfer Configurations	7-17
Transfer using a Shared Directory (recommended)	7-17
FTP based Transfer	7-20
Execution Modes	7-22
Controlling ABAP Uploading / ABAP code in production	7-22
Managing ODI SAP Transport Requests	7-24
SAP Packages and SAP Function Groups	7-25

Log Files	7-26
Limitation of the SAP BW Adapter	7-26

8 SAP ABAP ERP

Introduction	8-1
Concepts	8-1
Knowledge Modules	8-1
Overview of the SAP ABAP Integration Process	8-2
Reverse-Engineering Process	8-2
Integration Process	8-2
Installation and Configuration	8-3
System Requirements and Certification	8-3
Technology Specific Requirements	8-3
Connectivity Requirements	8-4
Installing and Configuring JCo	8-5
Installing ODI SAP Components into SAP System	8-6
Requesting FTP Server Access	8-6
Validating the SAP Environment Setup	8-6
Gathering SAP Connection Information	8-7
Gathering FTP Connection Information	8-8
Gathering Shared Directory Information	8-8
Adding the Open Tool	8-9
Defining the Topology	8-9
Create the File Data Server	8-10
Create a File Data Server	8-10
Create the File Schema	8-10
Create the SAP ABAP Data Server	8-11
Create the SAP ABAP Data Server	8-11
Create the SAP ABAP Schema	8-14
Setting up the Project	8-14
Creating and Reverse-Engineering a Model	8-14
Creating a SAP ERP Model	8-14
Reverse-Engineering a SAP ERP Model	8-14
Designing a Mapping	8-15
Considerations for SAP ERP Integration	8-15
File Transfer Configurations	8-15
Transfer using a Shared Directory (recommended)	8-16
FTP based Transfer	8-18
Execution Modes	8-20
Controlling ABAP Uploading / ABAP code in production	8-20

Managing ODI SAP Transport Requests	8-22
SAP Packages and SAP Function Groups	8-23
Log Files	8-24
Limitations of the SAP ABAP Adapter	8-24

A Additional Information for SAP ABAP BW Adapter

SAP ABAP BW Required Privileges	A-1
Important points to consider	A-2
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 & Foreground)	A-4
Authorizations Required for LKM SAP BW Execution (Background & Foreground)	A-6
SAP Stand-Alone Connection Test	A-7
SAP Stand-Alone Connection Test	A-7

B Additional Information for SAP ABAP ERP Adapter

SAP ABAP ERP Required Privileges	B-1
Important points to consider	B-2
Authorizations Required for RKM SAP ERP Upload	B-2
Authorizations Required for RKM SAP ERP Execution	B-3
Authorizations Required for LKM SAP ERP Upload	B-4
Authorizations Required for LKM SAP ERP Execution	B-5
SAP Connection Test	B-5
SAP Stand-Alone Connection Test	B-6

C Components, Compatibility, and Upgrading of the ODI SAP Adapter

Components	C-1
Compatibility	C-1
Upgrading the ODI SAP Adapter	C-2
Method 1: Add as new KMs	C-3
Replace existing KMs	C-3
Upgrading from ODI 11g to ODI 12c	C-3

D Installing ODI SAP Components

Updating ODI SAP Components	D-1
Installing ODI SAP Components	D-1
Installing SAP Transport Request (TR)	D-2

Downloading the Transport Request files	D-2
Installing the Transport Request Files	D-2
Creating Transport Request for ODI SAP Production Objects	D-6
Creating Transport Request for ODI SAP Development Objects	D-8
Installing and Assigning SAP User Profile	D-9
Validating the ODI SAP Setup	D-11
Validating the Shared Folder Setup	D-12
Validating the FTP Setup	D-12
Validating SAP Privileges	D-13
Validating SAP Transport Layer Name	D-14
Uninstalling ODI SAP Components	D-14
Validating Uninstallation of ODI SAP Components	D-16

E Moving ODI and SAP Components from Development to Production

Transport Request (TR)	E-1
Viewing the List of TRs	E-2
Description of TR Fields	E-3
Transport Request (TR) relevant to DEV	E-3
Transport Request (TR) relevant to PROD	E-4
Remote Function Call (RFC)	E-4
Adding Mappings Under Same Transport Request	E-6
KM Options Descriptions	E-8
Creating ODI Package	E-9
Generating ODI Scenario	E-11
Moving ODI Scenario to production repository (optional)	E-11
Releasing ODI SAP Components TR into Production	E-11
Transports are created after TR import	E-12
Transports are created using RKM process	E-13
Releasing the ODI Mapping TR into Production	E-14
Executing the ODI Scenario	E-16

Preface

This manual describes how to configure and work with the Application Adapters in Oracle Data Integrator.

This preface contains the following topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This document is intended for developers and administrators who want to use Oracle Data Integrator Application Adapters as a development tool for their integration processes.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

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

- Release Notes for Oracle Data Integrator
- Understanding Oracle Data Integrator
- Administering Oracle Data Integrator
- Developing Integration Projects with Oracle Data Integrator
- Installing and Configuring Oracle Data Integrator
- Upgrading Oracle Data Integrator

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

Conventions

The following text conventions are used in this document:

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

1

Introduction

Oracle Data Integrator uses Application Adapters to integrate data from and to enterprise applications. You can obtain information on the terminology used in this guide and how to use the guide in this section.

This book describes how to configure and work with the Application Adapters in Oracle Data Integrator for the following technologies:

- [Oracle E-Business Suite](#)
- [JD Edwards World](#)
- [Oracle PeopleSoft](#)
- [SAP ABAP BW](#)
- [SAP ABAP ERP](#)

This chapter includes the following sections:

- [Terminology](#)
- [Using This Guide](#)

Terminology

This section defines some common terms that are used in this document and throughout the related documents mentioned in the [Preface](#).

Knowledge Module

Knowledge Modules (KMs) are components of Oracle Data Integrator's Open Connector technology. KMs contain the knowledge required by Oracle Data Integrator to perform a specific set of tasks against a specific technology or set of technologies.

Combined with a connectivity layer such as, for example, JDBC, JMS, or JCA, Knowledge Modules define an Open Connector that performs defined tasks against a technology, such as connecting to this technology, extracting data from it, transforming the data, checking it, integrating it, etc.

Application Adapter

An *adapter* is a group of Knowledge Modules. In some cases, this group also contains an attached technology definition for Oracle Data Integrator.

Oracle Application Adapters for Data Integration provide specific software components for reverse-engineering metadata from, and extracting bulk data from various applications.

Using This Guide

This guide provides conceptual information and processes for working with application adapters and technologies supported in Oracle Data Integrator.

Each chapter explains how to configure a given technology, set up a project and use the technology-specific application adapters to perform integration operations.

Some knowledge modules are not technology-specific and require a technology that support an industry standard. These knowledge modules are referred to as *Generic* knowledge modules and are designed to work respectively with any ANSI SQL-92 compliant database and any JMS compliant message provider. See Generic SQL in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for more information.

Add link to JMS chapter along with Generic SQL .

When these generic knowledge module can be used with a technology, the technology chapter will mention it. However, it is recommended to use technology-specific knowledge modules for better performances and enhanced technology-specific feature coverage.

Before using a knowledge module, it is recommended to review the knowledge module description in Oracle Data Integrator Studio for usage details, limitations and requirements. In addition, although knowledge modules options are pre-configured with default values to work out of the box, it is also recommended to review these options and their description.

The chapters in this guide will provide you with the important usage, options, limitation and requirement information attached to the technologies and application adapters.

2

Oracle E-Business Suite

Oracle Data Integrator provides specific Knowledge Modules (KMs) for handling E-Business Suite data. You can obtain information on how to work with these KMs in this section.

This chapter includes the following sections:

- [Introduction](#)
- [Installation and Configuration](#)
- [Setting Up the Topology](#)
- [Setting up an Integration Project](#)
- [Creating an Oracle Model and Reverse-Engineering E-Business Suite Tables](#)
- [Designing a Mapping](#)

Introduction

Oracle E-Business Suite (EBS) is a suite of integrated software applications that provides a complete solution to the business needs of Oracle customers.

Concepts

The EBS Knowledge Modules provide support for the following capabilities:

- Reverse-engineering EBS objects: RKM E-Business Suite can be used to reverse-engineer E-Business Suite data structures.
- Data extraction from EBS: Standard Oracle or SQL LKMs can be used to extract data from E-Business suite using objects such as Tables, Views, and KeyFlexfields.
- Data integration to EBS: IKM E-Business Suite can be used to integrate data to E-Business Suite using Open Interface tables. The "Open Interface" API encapsulates several Oracle-specific interfaces and ensures data integrity. An Open Interface is made up of:
 - Several Interface tables to be loaded. These tables are the incoming data entry points for E-Business Suite.
 - Several programs that validate and process the insertion of the data from the interface tables into E-Business Suite.

Oracle Data Integrator Knowledge Modules for Oracle E-Business Suite interact with the database tier to extract metadata and load data. While loading data, it also interacts with the Concurrent Processing Server of the application tier.

Knowledge Modules

Oracle Data Integrator provides the Knowledge Modules listed in [Table 2-1](#) for handling E-Business Suite data. These specific EBS KMs provide comprehensive, bidirectional connectivity between Oracle Data Integrator and E-Business Suite, which enables you to extract and load data. The Knowledge Modules support all modules of E-Business Suite and provide bidirectional connectivity through EBS objects tables/ views and interface tables.

Table 2-1 E-Business Suite KMs

Knowledge Module	Description
IKM E-Business Suite (Open Interface)	<p>The IKM E-Business Suite is used to load data to EBS interface tables and submit Concurrent request (which loads from interface tables to base tables).</p> <p>This Integration Knowledge Module:</p> <ul style="list-style-type: none"> Integrates data from any source to Interface Tables in incremental update mode. Enables data control: invalid data is isolated in the Error Table and can be recycled. <p>In addition to loading the interface tables, it provides the following optional actions:</p> <ul style="list-style-type: none"> Create a Group ID for the first mapping in a batch. Use this Group ID in subsequent mappings. Delete this Group ID when loading the last table in the batch. Execute an Open Interface program if at any point in a batch it is required to call an E-Business Suite Interface program and once all required interface tables have been loaded. <p>Note that the IKM E-Business Suite (Open Interface) KM must only be used to load interface tables. Writing directly in the E-Business Suite physical tables is not supported.</p>
RKM E-Business Suite	<p>This KM reverse-engineers E-Business Suite data structures. It reverses EBS objects such as tables, views, flexfields and interface-tables structures in E-Business Suite (columns, primary keys and foreign keys).</p>

Installation and Configuration

Make sure you have read the information in this section before you start working with the E-Business Suite data:

- [System Requirements and Certifications](#)
- [Technology Specific Requirements](#)
- [Connectivity Requirements](#)

System Requirements and Certifications

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 list of supported platforms and versions is available on Oracle Technical Network (OTN):

<http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html>.

Technology Specific Requirements

There are no technology-specific requirements for using E-Business Suite data in Oracle Data Integrator.

Connectivity Requirements

There are no connectivity requirements for using E-Business Suite data in Oracle Data Integrator.

Setting Up the Topology

This step consists in declaring in Oracle Data Integrator the data server, physical schema and logical schema for the Oracle database that stores the E-Business Suite data.

Create an Oracle Data Server

Create a data server for the Oracle technology as described in *Creating an Oracle Data Server* in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*. This data server must point to the Oracle database instance that stores the E-Business Suite data.

Create an Oracle Physical Schema

Create an Oracle physical schema using the standard procedure, as described in *Creating a Physical Schema* in *Administering Oracle Data Integrator*. This schema must point to the Oracle schema that contains the synonyms pointing to the E-Business Suite tables.

 **Note:**

The physical schema must represent the Oracle schema containing the synonyms pointing to the E-Business Suite tables. This schema is usually called APPS. It must not point directly to the Oracle schemas containing the Application physical tables. These are usually named after the related applications.

Note also that for reverse-engineering, the Oracle user specified in the data server to which the Physical Schema is attached, must have the privileges to select from APPLSYS tables and the Oracle Data dictionary.

Create for this physical schema a logical schema using the standard procedure, as described in *Creating a Logical Schema* in *Administering Oracle Data Integrator* and associate it in a given context.

 **Note:**

The Oracle schema containing the E-Business Suite tables and the Oracle schema containing the synonyms that point to these tables should never be defined as a Work Schema in a physical schema definition. Moreover, these Oracle schemas must not be used as staging area for a mapping.

Setting up an Integration Project

Setting up a project using E-Business Suite features follows the standard procedure. See *Creating an Integration Project* of the *Developing Integration Projects with Oracle Data Integrator*.

Import the following KMs into your Oracle Data Integrator project:

- IKM E-Business Suite (Open Interface)
- RKM E-Business Suite

In addition to these specific EBS KMs, import the standard Oracle LKMs and CKMs to perform data extraction and data quality checks with an Oracle database. See Oracle Database in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for a list of available KMs.

Creating an Oracle Model and Reverse-Engineering E-Business Suite Tables

This section contains the following topics:

- [Create an Oracle Model](#)
- [Reverse-Engineer E-Business Suite Tables](#)

Create an Oracle Model

Create an Oracle Model based on the Oracle technology and on the logical schema created when configuring the E-Business Suite Connection using the standard procedure, as described in *Creating a Model* of the *Developing Integration Projects with Oracle Data Integrator*.

 **Note:**

There is no E-Business Suite technology defined in Oracle Data Integrator. The data model is created on top of the logical schema corresponding to the Oracle database hosting the EBS data.

Reverse-Engineer E-Business Suite Tables

The RKM E-Business Suite can reverse-engineer the installed E-Business Suite tables, enriching them with information retrieved from the E-Business Suite Integration repository.

The reverse-engineering process returns the following information:

- The installed E-Business Suite (Modules) as sub-models
- For each module sub-model, sub-models for Tables, Views, Flexfields, and Interface Tables
- The tables as datastores with their columns as attributes, and their constraints (Primary and Foreign Keys).
- Comments on the tables

To perform a Customized Reverse-Engineering of EBS tables with the RKM E-Business Suite, use the usual procedure, as described in Reverse-engineering a Model of the *Developing Integration Projects with Oracle Data Integrator*. This section details only the fields specific to EBS tables:

1. In the Reverse Engineer tab of the Oracle Model, select the RKM E-Business Suite.
2. Set the RKM options as follows:
 - Applications List: Enter the list of the applications' short name, for example INV.
 - Only Installed Applications: Set this option to YES to reverse-engineer only installed and shared applications. If this option is set to NO, all applications are reverse-engineered.
 - Min Rows: Leave the default value 0, if you want to reverse-engineer all the tables. If you want to reverse-engineer only tables with a minimum number of rows, specify in this option the minimum number of rows.
 - Description Mask: Specify the description mask for filtering the reverse-engineered objects based on their description in E-Business Suite.
 - Flexfields: If this option is set to YES, applications' flexfields are reverse-engineered.
 - Interface Tables: If this option is set to YES, applications' interface tables are reverse-engineered.
3. Specify the reverse mask in the Mask field in order to select the tables to reverse. The Mask field, in the Reverse Engineer tab, filters reverse-engineered objects based on their name.

 **Note:**

The Mask field and the Description Mask option are implemented using SQL Like. The patterns that you can choose from are:

- % the percentage symbol enables you to match any string of any length (including zero length)
- _ the underscore symbol enables you to match a single character

The reverse-engineering process returns the applications and tables as sub-models and datastores. You can use Oracle Applications as a source or a target of your mappings.

Example 2-1 Features of the E-Business Suite Reverse-Engineering

Reverse-engineering E-Business Suite Tables involves the following features:

- The E-Business Suite Modules are reversed as sub-models. The sub-model names correspond to the application names.
- Each application sub-model is divided into sub-models for Tables, Views, Flexfields and Interface Tables.
- The tables/views are reversed as datastores and their columns as attributes, along with their constraints (Primary and Foreign keys).
- A sub-model called *Flexfield on <AppName>* is created for each application. Datastores in the Flexfield sub-model correspond to *Concatenated_Segment_Views* of registered Key flexfields for the application. These objects are a subset of Views. The datastores in the Flexfields sub-folder are named after the flexfields.
- Datastores in Interface-Table sub-model correspond to tables whose names contain the pattern INTERFACE. These objects are a subset of tables.

 **Note:**

Some of the Open Interfaces (as specified in EBS Integration Repository) may have interface tables whose names may not contain the pattern INTERFACE in their names.

Example 2-2 Limitations of the E-Business Suite Reverse-Engineering Process

This section covers restrictions on reverse-engineering E-Business Suite Tables:

- Selective reverse-engineering cannot be used with this Knowledge Module.
- The Min Rows option requires Oracle statistics to be computed on all tables.
- If the Oracle user defined in the Oracle Data Integrator data server is not the owner of the tables to reverse-engineer, you must define synonyms for this user on these tables.
- Only KeyFlexfields are supported. Descriptive FlexFields are not supported.

Designing a Mapping

You can use E-Business Suite as a source and a target of a mapping.

The KM choice for a mapping determines the abilities and performance of this mapping. The recommendations in this section help in the selection of the KM for different situations concerning loading and integrating EBS data.

Loading Data from E-Business Suite

When using E-Business Suite as a source, you extract data from the Applications to integrate them into another system (Data warehouse, other database...).

Extracting data from E-Business Suite is performed with regular mappings sourcing from an Oracle Database. The knowledge modules working with the Oracle database technology can be used for this purpose. See Loading Data from Oracle in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for more information.

Integrating Data in E-Business Suite through the Open Interface

Oracle Data Integrator provides the IKM E-Business Suite (Open Interface) to integrate data in E-Business Suite. The integration process into E-Business Suite is as follows:

1. A set of Open Interface tables is loaded in a batch in a given transaction. This transaction is identified by a *Group ID*. Note the following concerning the Group ID:
 - For the first table in the batch, create a Group ID if it does not exist.
 - For the subsequent tables in the batch, use this Group ID when loading other tables in the batch.
 - When loading the last table in the batch, delete this Group ID.
2. If at any point in a batch it is required to call an E-Business Interface program, then you must validate and process data for the interface tables by executing an *Open Interface Program*. The batch is finalized by the Open Interface Program call that loads the base tables from the Open Interface tables.

These operations are supported by the IKM E-Business Suite (Open Interface). This IKM is used like the IKM Oracle Incremental Update and supports similar options to load the Open Interface tables. This section describes the options specific to Open Interfaces. See Oracle Database in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for more information about the IKM Oracle Incremental Update.

The configuration of mappings for actions specific to E-Business Suite, such as Group ID handling and the execution of Open Interface programs, is detailed in the [Managing Group IDs](#) and [Executing an Open Interface Program](#).

Managing Group IDs

A transaction that integrates data into E-Business Suite is a batch identified by its *Group ID*. For example, if you load several interface tables to create a product in E-

Business Suite, all of these loading operations as well as the calls to the validation and processing programs will use this batch's Group ID.

This section contains the following topics:

- [Creating a Group ID](#)
- [Using an existing Group ID](#)
- [Deleting an existing Group ID](#)

Creating a Group ID

You must force the creation of a Group ID in the first mapping that loads a group of interface tables in one single batch.

To create a Group ID in a mapping:

1. Set the following in the KM options:
 - Set `OA_CREATE_NEW_GROUP_ID` to `YES`
 - Provide a Group ID Name in the `OA_GROUP_ID_NAME` option.

Note:

The Group ID Name must be unique at a given instant. You must use the `OA_REMOVE_GROUP_ID` option to remove a Group ID at the end of the batch processing.

- Give a valid SQL expression for the Group ID value in the `OA_GROUP_ID_EXPRESSION` option. Use an Oracle Database sequence value, for example `<SEQUENCE_NAME>.NEXTVAL`
2. In the mapping, select the flag `UD1` for all the columns of the interface table that you want to load with the Group ID value and set the mapping value to `0`.

In the following mappings belonging to a batch, you must use an existing Group ID.

Using an existing Group ID

To use an existing Group ID in a mapping:

1. Set `OA_USE_EXISTING_GROUP_ID` IKM option to `Yes`.
2. Provide the Group ID Name in the `OA_GROUP_ID_NAME` IKM option.
3. In the mapping, select the flag `UD1` for all the columns that you want to load with the Group ID value and set the mapping value to `0`.

In the last mapping that loads a batch of interface tables, you may delete a Group ID that is no longer necessary.

Deleting an existing Group ID

To delete an existing Group ID:

1. Select the `OA_REMOVE_GROUP_ID` option.
2. Provide the Group ID Name in the `OA_GROUP_ID_NAME` option.

3. In the mapping, select the flag UD1 for all the columns of the interface table that you want to load with the Group ID value and set the mapping value to **0**.

 **Note:**

The Group IDs are stored in an SNP_OA_GROUP table that is created in the work schema specified in the physical schema that points to the Oracle Applications Interface tables. The Group ID is referenced in Oracle Data Integrator by a unique Group ID Name.

Executing an Open Interface Program

In Oracle Data Integrator mappings, when a set of interface tables is loaded, it is necessary to call an Open Interface program in order to validate and process the data in the E-Business Suite interface tables. You can use an existing Group ID in this call (see [Using an existing Group ID](#)), or create it (see [Creating a Group ID](#)) in the same mapping, if the Open Interface only contains a single table. The execution of the Open Interface program is started in the last mapping of a package. This mapping populates a set of Open Interface tables and usually deletes the Group ID, if no longer needed.

To execute an Open Interface Program:

1. Set the SUBMIT_PROGRAM option to YES.
2. Provide the name of the program to call in the OA_PROGRAM option.

 **Note:**

For a list of available Open Interface programs and their parameters, refer to the E-Business Suite module API and Open Interface documentation or the E-Business Suite Integration repository.

3. Specify the program parameters in the OA_ARGUMENTS option. The parameters are specified in the following format:

```
argument_name => 'argument value', argument_name => 'argument  
value' ...
```

If one argument must take the value of the Group ID, you must then specify
argument Name => v_group_id.

4. You must also specify the context parameters for the session that will execute the program by setting the values of the following options:
 - OA_USER_NAME: E-Business Suite User Name
 - OA_REPONSIBILITY: E-Business Suite Responsibility Name
 - OA_LANGUAGE: Language used for the responsibility
 - OA_APPLICATION: Application to which the responsibility belongs

3

JD Edwards EnterpriseOne

Oracle Data Integrator provides specific Knowledge Modules (KMs) for handling JDE EnterpriseOne data. You can obtain information on how to work with these KMs in this section.

This chapter includes the following sections:

- [Introduction](#)
- [Installation and Configuration](#)
- [Setting Up the Topology](#)
- [Setting up an Integration Project](#)
- [Creating and Reverse-Engineering a Model](#)
- [Designing a Mapping](#)

Introduction

JD Edwards (JDE) EnterpriseOne is an integrated applications suite of comprehensive ERP software that combines business value, standards-based technology, and deep industry experience into a business solution with a low total cost of ownership.

Concepts

The JDE Knowledge Modules for Oracle Data Integrator use mature database-level integration methods for JDE EnterpriseOne, in order to:

- Reverse-Engineer JDE EnterpriseOne data structures
- Read data from JDE EnterpriseOne (Direct Database Integration)
- Write data through the Z-tables to JDE Application (Interface Table Integration)

Knowledge Modules

Oracle Data Integrator provides the Knowledge Modules listed in the below table for handling JDE EnterpriseOne data. These specific JDE KMs provide connectivity and integration of the JDE EnterpriseOne platform with any database application through Oracle Data Integrator.

Table 3-1 JDE Knowledge Modules

Knowledge Module	Description
RKM JDE EnterpriseOne Oracle	Reverse-engineers the metadata of the applications' objects such as tables and interface tables from JDE EnterpriseOne installed on an Oracle database.

Table 3-1 (Cont.) JDE Knowledge Modules

Knowledge Module	Description
RKM JDE EnterpriseOne SQL Server	Reverse-engineers the metadata of the applications' objects such as tables and interface tables from JDE EnterpriseOne installed on SQL Server.
RKM JDE EnterpriseOne DB2 UDB	Reverse-engineers the metadata of the applications' objects such as tables and interface tables from JDE EnterpriseOne installed on IBM DB2 UDB database.
RKM JDE EnterpriseOne DB2 AS400	Reverse-engineers the metadata of the applications' objects such as tables and interface tables from JDE EnterpriseOne installed on IBM DB2 for iSeries server.
IKM JDE EnterpriseOne Control Append (UBE)	<p>Integrates data from any source to JDE EnterpriseOne. Integrates data in EnterpriseOne Z-table in control append mode.</p> <ul style="list-style-type: none"> • Data can be controlled: invalid data is isolated in the Error Table and can be recycled • The KM performs integration into JDE EnterpriseOne with a RunUBE batch command

Installation and Configuration

Make sure you have read the information in this section before you start working with the JDE EnterpriseOne data:

- [System Requirements and Certifications](#)
- [Technology Specific Requirements](#)
- [Connectivity Requirements](#)

System Requirements and Certifications

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 list of supported platforms and versions is available on Oracle Technical Network (OTN):

<http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html>

Technology Specific Requirements

In order to use the IKM JDE EnterpriseOne Control Append (UBE), the Oracle Data Integrator run-time agent must be installed on the JDE Server where the RunUBE utility is installed.

In order to use the RKM JDE EnterpriseOne DB2 UDB to reverse-engineer tables and Z-tables, the IBM DB2 UDB database should be able to access data stored in different DB2 databases. The following steps describe how to configure the access to DB2 family data sources:

1. Set up and configure the federated server and database. Configuring the federated server to access DB2 data sources involves supplying the server with information about the DB2 data sources and objects you want to access. You can configure access to DB2 data sources in two ways:

- Through the DB2 Control Center
- Through the DB2 Command Center or command line processor (CLP)

2. Add a DB2 data source to a federated server:

- a. Catalog a node entry in the federated node directory.

For example, if TCP/IP is your communication protocol, issue the following command:

```
CATALOG TCPIP NODE <db2node> REMOTE <system> SERVER <server_name>
```

- b. Catalog the remote database in the federated system database director using the following command:

```
CATALOG DATABASE <db_name> AS <alias_name> AT NODE <db2_node>  
AUTHENTICATION SERVER
```

- c. Create the wrapper using the following command:

```
CREATE WRAPPER DRDA
```

DRDA is the default wrapper name to access the DB2 family of products. Every DB2 Server Edition (Enterprise, Personal, Workgroup) includes the DRDA wrapper.

- d. Create the server definition.

```
CREATE SERVER <server_name> TYPE <type> VERSION <version> WRAPPER  
<wrapper_name> AUTHORIZATION <user> PASSWORD <password> OPTIONS  
(DBNAME <db_name>)
```

where

```
AUTHORIZATION <user>
```

Is the authorization ID at the data source. This ID must have BINDADD authority at the data source. This value is case-sensitive.

```
PASSWORD <password>
```

Is the password associated with the authorization ID at the data source. This value is case-sensitive.

```
DBNAME <db_name>
```

Is the alias for the DB2 database that you want to access. You defined this alias when you cataloged the database using the CATALOG DATABASE command. This value is case-sensitive.

Although the database name is specified as an option in the CREATE SERVER statement, it is required for DB2 data sources.

- e. Create the user mappings. If a user's authorization ID to access the federated database differs from the user's authorization ID to access a data source, you need to define a user mapping between the two authorization IDs.

```
CREATE USER MAPPING FOR <db2user> SERVER <server_name> OPTIONS
(REMOTE_AUTHID <remote_user> REMOTE PASSWORD <remote_password>)
```

Note that REMOTE_AUTHID is the connect authorization ID at the DB2 family data source server to which you are mapping the db2user, not the bind authorization ID.

- f. Test the connection to the DB2 data source server.
- g. Create the nicknames for every table object that resides on the different database that you want to access. The federated database relies on catalog statistics for nicknamed objects to optimize query processing. Create a nickname for each data source object using the following command:

```
CREATE NICKNAME <nickname> FOR
<server_name>.<schema_name>.<table_name>
```

Create the nicknames as described in the table below.

Table 3-2 Nicknames for Table Objects

Nickname	Table Name	Schema
F9860	F9860	JDE_OBJECT_LIBRARIAN
F98710	F98710	JDE_CENTRAL_OBJECTS
F98711	F98711	JDE_CENTRAL_OBJECTS
F98712	F98712	JDE_CENTRAL_OBJECTS
F98713	F98713	JDE_CENTRAL_OBJECTS
F9802	F9802	JDE_DATA_DICTIONARY

 **Note:**

Even if the table names and schemas might be different between JDE version 8 and 9 and differ from the values listed in the above table, the nicknames should stay the same.

See the *IBM DB2 Universal Database - Federated Systems Guide* for more information.

Connectivity Requirements

Oracle Data Integrator connects to the database hosting the JDE data using JDBC connectivity. For detailed information on JDBC connectivity with Oracle database, Microsoft SQL Server, IBM DB2 UDB, and IBM DB2 for iSeries, see the following sections in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*:

- Oracle Database Connectivity Requirements
- Microsoft SQL Server Connectivity Requirements
- IBM DB2 for iSeries Connectivity Requirements

- IBM DB2 UDB Specific Requirements

Setting Up the Topology

This step consists of declaring in Oracle Data Integrator the data server, physical schema and logical schema that will be used to store the JDE data.

Create a Data Server

Depending on the underlying technology, the JDE tables can be stored in an Oracle schema, a Microsoft SQL Server database, an IBM DB2 UDB schema or in an IBM DB2 for iSeries library.

Create a data server for the technology hosting the JDE tables. For more information, see the following sections in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*:

- Creating an Oracle Data Server
- Creating a Microsoft SQL Server Data Server
- Creating a DB2/400 Data Server
- IBM DB2 UDB Specific Requirements

This data server must point to the instance, schema, database, or library (in the subsequent sections, the term schema will be used for all technologies) that stores the JDE data.

Create a Physical Schema

Create a physical schema under the data server that you have created in [Create a Data Server](#). Use the standard procedure, as described in Creating a Physical Schema in *Administering Oracle Data Integrator*.

This schema must point to the schema that contains the JDE tables that you want to reverse-engineer.

 **Note:**

The schema storing the JDE tables should never be defined as a work schema in the physical schema definition. Moreover, this schema must not be used as staging area of a mapping.

Create for this physical schema a logical schema using the standard procedure, as described in Creating a Logical Schema in *Administering Oracle Data Integrator* and associate it in a given context.

Setting up an Integration Project

Setting up a project using JDE features follows the standard procedure. See Creating an Integration Project in *Developing Integration Projects with Oracle Data Integrator*.

Import the following KMs into your Oracle Data Integrator project:

- IKM JDE EnterpriseOne Control Append (UBE)
- Depending on the technology hosting your JDE tables, import one of the following:
 - RKM JDE EnterpriseOne Oracle
 - RKM JDE EnterpriseOne SQL Server
 - RKM JDE EnterpriseOne DB2 UDB
 - RKM JDE EnterpriseOne DB2 AS400

In addition to these specific JDE KMs, import the standard LKMs for the technology hosting your JDE tables. For a list of available KMs, see the following sections in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*:

- Oracle Database Knowledge Modules
- Microsoft SQL Server Knowledge Modules
- IBM DB2 for iSeries Knowledge Modules
- IBM DB2 UDB Knowledge Modules

Creating and Reverse-Engineering a Model

This section contains the following topics:

- [Create a Model](#)
- [Reverse-Engineer JDE Tables](#)

Create a Model

Create a model based on the technology hosting the JDE tables and on the logical schema created when configuring the JDE Connection using the standard procedure, as described in *Creating a Model in Developing Integration Projects with Oracle Data Integrator*.

Note:

There is no JDE EnterpriseOne technology defined in Oracle Data Integrator. The data model is created with the logical schema corresponding to the Oracle database hosting the JDE data.

Reverse-Engineer JDE Tables

The JDE RKMs are able to reverse-engineer JDE tables. These RKMs retrieve metadata from JDE objects such as tables and interface tables.

To perform a Customized Reverse-Engineering of JDE tables with the JDE RKMs, use the usual procedure, as described in *Reverse-engineering a Model in Developing Integration Projects with Oracle Data Integrator*. This section details only the fields specific to JDE tables:

1. In the Reverse tab of the Model, select the RKM JDE EnterpriseOne <database>. In this chapter, <database> refers to the technology containing the JDE data.
2. Set the RKM options as follows:
 - JDE_CENTRAL_OBJECTS: Specify the Oracle Schema or Microsoft SQL Server Database storing the JDE Central objects
 - JDE_DATA_DICTIONARY: Specify the Oracle Schema or Microsoft SQL Server Database storing the JDE data dictionary
 - JDE_OBJECT_LIBRARIAN: Specify the Oracle Schema or Microsoft SQL Server Database storing the JDE Object librarian
 - JDE_CONTROL_TABLES: Specify the Control Tables schema

 **Note:**

To find the schema required in the options JDE_CENTRAL_OBJECTS, JDE_DATA_DICTIONARY, JDE_OBJECT_LIBRARIAN, and JDE_CONTROL_TABLES, you can either ask your application manager or query the table F98611(Data Source Master).

- JDE_DATA_TABLES: Set this option to YES to reverse-engineer data tables
- JDE_Z_TABLES: Set this option to YES to reverse-engineer interface tables (Z-tables)
- JDE_MODULES: Indicate the JDE System Short Name, for example: 00 for Foundation Environment, 01 for Address Book, and 02 for Electronic Mail

 **Note:**

You can also specify a list of modules. In the list, the modules must be separated by commas and enclosed within single-quote characters, for example:

```
'01', '02', '04'
```

- JDE_LANGUAGE: Indicate the language used for retrieving object descriptions and comments, for example E for English, F for French, and S for Spanish
3. Specify the reverse mask in the Mask field in order to select the tables to reverse. The Mask field, in the Reverse tab, filters reverse-engineered objects based on their name. The Mask field must not be empty and must contain at least the percentage symbol (%).

The reverse-engineering process returns the datastores grouped per module. You can use these datastores as a source or a target of your mappings.

Designing a Mapping

You can use JDE data tables as a source of a mapping. JDE Z-tables can be used as the target of a mapping.

The KM choice for a mapping determines the abilities and performance of this mapping. The recommendations in this section help in the selection of the KM for different situations concerning loading and integrating JDE data.

Loading Data from JDE EnterpriseOne

After performing a reverse-engineering using the RKM JDE EnterpriseOne <database>, you can use JDE data tables as a source of a mapping to extract data from the JDE application and integrate them into another system (Data warehouse, other database and so forth).

Using JDE EnterpriseOne as a source in these conditions is the same as using an Oracle, Microsoft SQL Server, DB2/400 or an IBM DB2 UDB datastore as a source in a mapping. The generic SQL, Oracle Database, or Microsoft SQL Server, IBM DB2 for iSeries, and IBM DB2 UDB KMs can be used for this purpose. See the following chapters in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for more information:

- Oracle Database
- Generic SQL
- Microsoft SQL Server
- IBM DB2 for iSeries
- IBM DB2 UDB

Integrating Data in JDE EnterpriseOne

After performing a reverse-engineering using the RKM JDE EnterpriseOne <database>, you can use JDE Z-tables as a target of a mapping to load data from any system to the JDE application with the IKM JDE EnterpriseOne Control Append (UBE).

The integration of data into JDE EnterpriseOne is performed in two phases:

- During the first phase, data is integrated into a set of Z-tables using several mappings, without calling the RunUBE command. These mappings can use the IKM JDE EnterpriseOne Control Append (UBE) with the JDE_RUNUBE option set to `No`.
- During the second phase, the RunUBE command is launched to integrate the data from these Z-tables into JDE EnterpriseOne. This is typically done in the mapping loading the last required Z-table. This mapping also uses the IKM JDE EnterpriseOne Control Append (UBE) with the JDE_RUNUBE option set to `Yes`.

These mappings should be sequenced in a package.

Oracle Data Integrator can automatically call the RunUBE command to write to JDE. The RunUBE call should be activated in the IKM only after loading all the required Z-table for populating JDE. The capability to load the Z-Tables and the call of the RunUBE command is provided by the IKM JDE EnterpriseOne Control Append (UBE).

To create a mapping targeting JDE:

1. Create a mapping with Z-tables as target datastores.
2. Create joins, filters, and mappings as usual.
3. In the Flow tab, select the **IKM JDE EnterpriseOne Control Append (UBE)**.

4. Set the standard KM options (INSERT, COMMIT, FLOW_CONTROL).
5. If this mapping launches the RunUBE command, specify the KM options as follows:
 - a. Set the JDE_RUNUBE option to Yes.
 - b. Specify the JDE_DIRECTORY in which the RunUBE command is executed.
 - c. If you want to create a password file, set the password related options as shown in the below table.

Table 3-3 Password Related KM Options

Option	Value	Notes
JDE_CREATE_PWD_FILE	Yes	To enhance RunUBE security in a UNIX or iSeries environment, when the RunUBE command is submitted, the system reads the text file specified in the JDE_PWD_FILE option and uses the JD Edwards EnterpriseOne user ID and password as indicated in the text file.
JDE_PWD_FILE	Absolute path of the password security file	This file contains the user id and password specified in the JDE_USER_ID and JDE_PWD options.
JDE_DELETE_PWD_FILE	E D	Enter D to delete the password file. Enter F to keep the password file. Note that even if the password file is removed after the execution of the command, the file should be kept in a secure location on the server file system.
JDE_USER_ID	JDE EnterpriseOne user ID	The user must have permissions to run the report.
JDE_PWD	JDE EnterpriseOne password	The EnterpriseOne password that corresponds to the user ID.

- d. Set the parameters for the RunUBE command as shown in the below table.

Table 3-4 RunUBE Command Related KM Options

Option	Value	Notes
JDE_ENVIRONMENT	The JDE EnterpriseOne environment	

Table 3-4 (Cont.) RunUBE Command Related KM Options

Option	Value	Notes
JDE_ROLE	The JDE EnterpriseOne role	
JDE_REPORT	The system name of the report that you want to process	For example: APS Outbound Processor (R34A400) and APS Inbound Processor (R34A410) for flat files, and the APS SCBM 2.0 Outbound Processor (R34A700) and APS Master Inbound Processor (R34A820) for XML files
JDE_VERSION	The name of the version of the report that you want to process	For example: XJDE0001. Note that you must enter a version name; you cannot submit the template of a report.
JDE_JOB_QUEUE	The name of the job queue to which the system should route the batch job	For example: QBATCH
JDE_PROCESSING_MODE	The processing mode	Enter B to use batch processing. In this case, the system uses the Job Control Status Master table (F986110) to assign the report a place in the queue. Enter I to use the interactive mode. This mode runs the report immediately outside of the JDE EnterpriseOne queuing mechanism.
JDE_HOLD_CODE	The hold code	Enter P to send the output to a printer immediately after the job completes. Enter H to hold the processed file without printing. You can print the job later using the Work With Servers program (P986116) that is accessible from the System Administration Tools menu (GH9011).
JDE_SAVE_CODE	The save code	Enter S to save the file after processing is complete. The delete option (D) is reserved for future use. Currently, the delete option is disabled.

Limitations of the IKM JDE EnterpriseOne Control Append (UBE)

- The TRUNCATE option cannot work if the target table is referenced by another table (foreign key).
- When using the RECYCLE_ERRORS option, an Update Key must be set for your mapping.
- When using this module with a journalized source table, data is automatically filtered to not include source deletions.
- The FLOW_CONTROL and STATIC_CONTROL options call the Check Knowledge Module to isolate invalid data (if no CKM is set, an error occurs). Both options must be set to No when a mapping populates a TEMPORARY target datastore.
- The RunUBE command must be executed on the JDE server.
- The Oracle Data Integrator run-time agent must be installed on this server.
- Besides the information whether the RunUBE command has been started or not, the RunUBE command does not give any further details about the execution of the program. To know more about the execution of the program you can either view the log file created by the JDE server or connect to your JDE application and look for the application View Job Status (Application = P986110, Form = W986116A).

4

JD Edwards World

Oracle Data Integrator provides specific Knowledge Modules (KMs) for handling JDE World data. You can obtain information on how to work with these KMs in this section. This chapter includes the following sections:

- [Introduction](#)
- [Installation and Configuration](#)
- [Setting Up the Topology](#)
- [Setting up an Integration Project](#)
- [Creating and Reverse-Engineering a Model](#)
- [Designing a Mapping](#)

Introduction

JD Edwards (JDE) World is an integrated applications suite of comprehensive ERP software that combines business value, standards-based technology, and deep industry experience into a business solution with a low total cost of ownership.

Concepts

The JDE Knowledge Modules for Oracle Data Integrator use mature database-level integration methods for JDE World, in order to:

- Reverse-Engineer JDE World data structures
- Read data from JDE World (Direct Database Integration)
- Write data through the Z-tables to a JDE World Application (Interface Table Integration)

Knowledge Modules

Oracle Data Integrator provides the Knowledge Modules listed in [Table 4-1](#) for handling JDE World data. These specific JDE World KMs provide connectivity and integration of the JDE World platform with any database application through Oracle Data Integrator.

Table 4-1 JDE Knowledge Modules

Knowledge Module	Description
RKM JDE World	Reverse-engineers the metadata of the applications' objects such as tables and interface tables from JDE World installed on DB2 iSeries database, through DB2 iSeries JDBC driver (jt400).

Table 4-1 (Cont.) JDE Knowledge Modules

Knowledge Module	Description
RKM JDE World (JDE World JDBC Driver)	Reverse-engineers the metadata of the applications' objects such as tables and interface tables from JDE World installed on DB2 iSeries database, through JDE World JDBC Driver.
IKM JDE World Control Append	Integrates data from any source to JDE World. Integrates data in a Z-table in control append mode. <ul style="list-style-type: none"> Data can be controlled: invalid data is isolated in the Error Table and can be recycled The KM performs integration into JDE World with a RPG program

Installation and Configuration

Make sure you have read the information in this section before you start working with the JDE World data:

- [System Requirements and Certifications](#)
- [Technology Specific Requirements](#)
- [Connectivity Requirements](#)

System Requirements and Certifications

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 list of supported platforms and versions is available on Oracle Technical Network (OTN):

<http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html>.

Technology Specific Requirements

This section lists the technology specific requirements of the JDE World Knowledge Modules.

RKM JDE World

- `jt400.jar` - This jar file must be in the `~/ .odi/oracledi/userlib` folder.

RKM JDE World (JDE World JDBC Driver)

- `JDEWorldJDBC.jar` - This JDE World JDBC driver file must be in the `~/ .odi/oracledi/userlib` folder.

IKM JDE World Control Append

The RPG program requires the following files:

- `JDEWorldJDBC.jar` - This JDE World JDBC driver file must be in the `~/.odi/oracledi/userlib` folder.

See [Add Additional Drivers and Open Tools in the *Installing and Configuring Oracle Data Integrator*](#) for more information about these folders.

- `BaseJar.jar` - This jar file must be in the `~/.odi/oracledi/userlib` folder.
- `config.xml` - This configuration file must be in the `ODI_HOME/odi/studio/bin` folder.

 **Note:**

These three files are delivered with ODI and are located in the `ODI_HOME/odi/misc/jde-world` directory.

 **Tip:**

The `ODI_HOME/odi/misc/jde-world` directory also contains a `jde.properties` file. This properties file is a template that you can make use of if you are using the `JDE_SECURITY_FILE` option in the IKM JDE World Control Append. See [Integrating Data in JDE](#) for more information about this option. When using this template make sure to:

- Rename the properties file
- Enter the connection information (JD Edwards World user, password, environment, and address) .

See [Table 4-2](#) for more information about the connection related options. Also, an example of a security file is provided below the table.

- Move it to a directory that is accessible only by Oracle Data Integrator Studio or the standalone agent. Ensure that this directory is not accessible to any other user as the properties file contains the user name and password.

Connectivity Requirements

Oracle Data Integrator connects to the database hosting the JDE World data using JDBC connectivity.

The RKM JDE World (JDE World JDBC Driver) uses the JDE World JDBC driver to access the database to extract metadata information from JDE World. The JDE World JDBC driver provides the ability for non-JDE World applications to access JDE World data while maintaining the level of security and the flexibility built into the JDE World software.

Both the IKM JDE World Control Append and the RKM JDE World use the standard IBM Toolbox for Java driver (`jt400.jar`), which runs SQL queries to insert or access the database to extract metadata information for JDE World.

 **Note:**

in the JDBC URL, use the *SQL naming convention. Do not specify the naming convention to be `system as`, for example in :

```
jdbc:as400://192.0.2.1;translate binary=true;naming=system.
```

*SQL should always be used unless your application is specifically designed for *SYS. Oracle Data Integrator uses the *SQL naming convention by default.

For detailed information on JDBC connectivity with IBM DB2 for iSeries, see IBM DB2 for iSeries Connectivity Requirement in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*.

Setting Up the Topology

This step consists in declaring in Oracle Data Integrator the data server, physical schema and logical schema that will be used to store the JDE World data.

Create a Data Server

The JDE World tables are stored in an IBM DB2 for iSeries library.

When working with RKM JDE World:

Create a data server for the IBM DB2 for iSeries technology using the standard procedure, as described in Creating a DB2/400 Data Server of the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*.

 **Note:**

When defining the connection parameters for the data server, set in the JDBC URL field `translate binary=true`

For example:

```
jdbc:as400://192.0.2.1;translate binary=true
```

This data server must point to the library that stores the JDE World data.

When working with RKM JDE World (JDE World JDBC Driver):

Create a data server for the IBM DB2 for iSeries technology using following information:

- JDBC Driver: `com.jdedwards.as400.access.JDEWJDBCdriver`
- JDBC URL: `jdew://<host>;translatebinary=true;JDEWEnvironment=<name>;user=<name>;pwd=<password>;JDEWTableNomenclature=OBJN_OBJT;JDEWColumnNomenclature=FDFT_FDFN`

For options specified in the JDBC URL, refer to the *JDE World JDBC Driver User Guide*.

Create a Physical Schema

Create a physical schema under the data server that you have created in [Create a Data Server](#). Use the standard procedure, as described in Creating a Physical Schema in *Administering Oracle Data Integrator*.

This schema must point to the library that contains the JDE World tables that you want to reverse-engineer.

Note:

The library storing the JDE tables should never be defined as a work schema in the physical schema definition. Moreover, this library must not be used as staging area of a mapping.

Create for this physical schema a logical schema using the standard procedure, as described in Creating a Logical Schema in *Administering Oracle Data Integrator* and associate it in a given context.

Setting up an Integration Project

Setting up a project using JDE World features follows the standard procedure. See Creating an Integration Project of the *Developing Integration Projects with Oracle Data Integrator*.

Import the following KMs into your Oracle Data Integrator project:

- IKM JDE World Control Append
- RKM JDE World
- RKM JDE World (JDE World JDBC Driver)

In addition to these specific JDE World KMs, import the standard LKMs for the technology hosting your JDE World tables. For a list of available KMs, see IBM DB2 for iSeries Knowledge Modules in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*.

Creating and Reverse-Engineering a Model

This section contains the following topics:

- [Create a Model](#)
- [Reverse-Engineer JDE Tables](#)

Create a Model

Create a Model based on the IBM DB2/400 technology hosting the JDE World tables and on the logical schema created when configuring the JDE World connection using

the standard procedure, as described in *Creating a Model of the Developing Integration Projects with Oracle Data Integrator*.

Reverse-Engineer JDE Tables

The JDE RKMs are able to reverse-engineer JDE tables. These RKMs retrieve metadata from JDE objects such as tables and interface tables.

To perform a Customized Reverse-Engineering of JDE tables with the RKM JDE World or RKM JDE World (JDE World JDBC Driver), use the usual procedure, as described in *Reverse-engineering a Model of the Developing Integration Projects with Oracle Data Integrator*. This section details only the fields specific to JDE World tables:

1. In the Reverse tab of the Model, select RKM JDE World or RKM JDE World (JDE World JDBC Driver).
2. Set the options for the selected RKM:

For RKM JDE World (JDE World JDBC Driver), set the options as follows:

- JDE_MODULES: Indicate the JDE System Short Name, for example:
 - 00 for Foundation Environment
 - 01 for Address Book
 - 02 for Electronic Mail.
 - % for all JDE SystemsDefault is 01.

Note:

You can also specify a list of modules. In the list, the modules must be separated by commas and enclosed within single-quote characters, for example:

```
'01', '02', '04'
```

For RKM JDE World, set the options as follows:

- JDE_DATA_TABLES: Set this option to YES to reverse-engineer data tables
- JDE_Z_TABLES: Set this option to YES to reverse-engineer interface tables (Z-tables)
- JDE_MODULES: Indicate the JDE System Short Name, for example:
 - 00 for Foundation Environment
 - 01 for Address Book
 - 02 for Electronic Mail.
 - % for all JDE SystemsDefault is 01.

 **Note:**

You can also specify a list of modules. In the list, the modules must be separated by commas and enclosed within single-quote characters, for example:

```
'01', '02', '04'
```

- **JDE_LANGUAGE:** Indicate the language used for retrieving object descriptions and comments, for example **E** for English, **F** for French, and **S** for Spanish. Default is **E**.
3. Specify the reverse mask in the Mask field in order to select the tables to reverse. The Mask field, in the Reverse tab, filters reverse-engineered objects based on their name. The Mask field must not be empty and must contain at least the percentage symbol (%).

The reverse-engineering process returns the datastores grouped per module. You can use these datastores as a source or a target of your mappings.

Designing a Mapping

You can use JDE World data tables as a source of a mapping. JDE Z-tables can be used as the target of a mapping.

The KM choice for a mapping determines the abilities and performance of this mapping. The recommendations in this section help in the selection of the KM for different situations concerning loading and integrating JDE World data.

Loading Data from JDE

After performing a reverse-engineering using the RKM JDE World or RKM JDE World (JDE World JDBC Driver), you can use JDE World data tables as a source of a mapping to extract data from the JDE World application and integrate them into another system (Data warehouse, other database and so forth).

Using JDE World as a source in these conditions is the same as using a DB2/400 as a source in a mapping. The generic SQL and IBM DB2 for iSeries KMs can be used for this purpose. See the following chapters in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for more information:

- Generic SQL
- IBM DB2 for iSeries

Integrating Data in JDE

After performing a reverse-engineering using the RKM JDE World, you can use JDE Z-tables as a target of a mapping to load data from any system to the JDE World application with the IKM JDE World Control Append.

The integration of data into JDE World is performed in two phases:

- During the first phase data is integrated into a set of Z-tables using several mappings, without calling the RPG program. These mappings can use the IKM JDE World Control Append with the JDE_INVOKE option set to `No`.
- During the second phase the RPG program is launched to integrate the data from these Z-tables into JDE World. This is typically done in the mapping loading the last required Z-table. This mapping also uses the IKM JDE World Control Append with the JDE_INVOKE option set to `Yes`.

These mappings should be sequenced in a package.

Oracle Data Integrator can automatically call the RPG program to write to JDE World. The RPG program call should be activated in the IKM only after loading all the required Z-table for populating JDE. The capability to load the Z-Tables and the call of the RPG program is provided by the IKM JDE World Control Append.

To create a mapping targeting JDE World:

1. Create a mapping with Z-tables as target datastores.
2. Create joins, filters, and mappings as usual.
3. In the Physical diagram properties, go to the Integration Knowledge Module tab and select the **IKM JDE World Control Append**.
4. Set the standard KM options (INSERT, COMMIT, FLOW_CONTROL).
5. If this mapping launches the RPG program, specify the KM options as follows:
 - a. Set the JDE_INVOKE option to `Yes`.
 - b. If you want to create a security file, set the connection related options as shown in [Table 4-2](#).

Table 4-2 Connection Related KM Options

Option	Value	Notes
USE_SECURITY_FILE	Yes	To enhance security when the RPG program is submitted, the system reads the text file specified in the JDE_SECURITY_FILE option and uses the JD Edwards World user, password, environment, and address as indicated in the text file.

Table 4-2 (Cont.) Connection Related KM Options

Option	Value	Notes
JDE_SECURITY_FILE	Absolute path of the connection security file	This file contains the JDE World user, password, environment, and address specified in the JDE_USER, JDE_PASSWORD, JDE_ENVIRONMENT, and JDE_ADDRESS options. See the security file example below for more information. Mandatory if USE_SECURITY_FILE is set to yes. In this case, there is no need to set valuse for JDE_USER, JDE_PASSWORD, JDE_ENVIRONMENT, and JDE_ADDRESS. You can use the template security file that is delivered with the files required for the RPG program. For more information, see the Tip in Technology Specific Requirements .
JDE_USER	JDE World user	The user must have the ONEWORLD profile.
JDE_PASSWORD	JDE World password	This user's JDE World password.
JDE_ENVIRONMENT	JDE World environment	The JDE World environment
JDE_ADDRESS	JDE World address	The IP or full address of the iSeries server

The following example shows a security file. Make sure to use the same syntax and key names in your security file.

```
Username=ODI
Password=password
Environment=JDEENV
Address=iseries.organization.com
```

- c. Set the parameters for the RPG program as shown in [Table 4-3](#).

Table 4-3 RPG Program related KM Options

Option	Value	Notes
JDE_PRGNAME	The name of the PRG program	For example: P01051Z
JDE_CLNAME	The type of the PRG program	For example: J01051Z

Table 4-3 (Cont.) RPG Program related KM Options

Option	Value	Notes
JDE_VERSION	The name of the version of the report that you want to process	For example: XJDE0006. Note: <ul style="list-style-type: none"> Enter the version name of the report to duplicate and process; you cannot submit the template of a report ODI creates a temporary version
JDE_KCO	The batch ID	For example: 1 Note that the batch ID allows to specify which rows are to be processed by the RPG program.
JDE_EDTN	The transaction ID	For example: 1 Note that transaction ID allows to group several rows in a single transaction. This is typically the case for header-detail tables. The headers are linked to the detail by the transaction id. This IKM does not provide truncation.

Limitations of the IKM JDE World Control Append

- When using the RECYCLE_ERRORS option, an Update Key must be set for your mapping.
- When using this module with a journalized source table, data is automatically filtered to not include source deletions.
- The FLOW_CONTROL and STATIC_CONTROL options call the Check Knowledge Module to isolate invalid data (if no CKM is set, an error occurs).
- The RPG program must be executed on the JDE World iSeries server.
- The Oracle Data Integrator run-time agent can be installed on this server. However, it is not necessary to install the run-time agent on the server to run the RPG program. The RPG program can be executed using a local agent.
- Besides the information whether the RPG program has been started or not, ODI does not give any further details about the execution of the program. To know more about the execution of the program you can view the log file created by the iSeries server and issue the Work with Spooled Files (WRKSPFLF) command.

5

Oracle PeopleSoft

Oracle Data Integrator provides specific Knowledge Modules (KMs) for handling PeopleSoft data. You can obtain information on how to work with these KMs in this section.

This chapter includes the following sections:

- [Introduction](#)
- [Installation and Configuration](#)
- [Setting up the Topology](#)
- [Setting up the Project](#)
- [Creating and Reverse-Engineering a Model](#)
- [Designing a Mapping](#)

Introduction

Oracle Data Integrator integrates data extracted from Oracle PeopleSoft applications. It supports reverse-engineering of PeopleSoft metadata as well as scalable data extraction from PeopleSoft tables.

Concepts

The Oracle Data Integrator KMs for PeopleSoft use mature integration methods for PeopleSoft, in order to:

- Reverse-Engineer PeopleSoft data structures (Business Objects, tables, views, columns, keys, and foreign keys) in the form of datastores. Oracle Data Integrator provides two specialized knowledge modules (KMs) for performing this operation for PeopleSoft instances hosted in Oracle and Microsoft SQL Server databases.
- Extract data from PeopleSoft using a data-level integration approach. Data is extracted from the database tables containing the PeopleSoft data.

Knowledge Modules

Oracle Data Integrator provides the Knowledge Modules listed in [Table 5-1](#) for handling PeopleSoft data. These specific Knowledge Modules for PeopleSoft provide integration and connectivity between Oracle Data Integrator and the PeopleSoft platform.

These KMs enable data-level integration for PeopleSoft: Data extraction is performed directly on the PeopleSoft Business Objects tables. This method is read-only.

Table 5-1 Oracle PeopleSoft Knowledge Modules

Knowledge Module	Description
RKM PeopleSoft ORACLE	Reverse-engineering knowledge module to retrieve the business objects, tables, views, columns, keys, and foreign keys from PeopleSoft. The database hosting the PeopleSoft tables is Oracle.
RKM PeopleSoft MSSQL	Reverse-engineering knowledge module to retrieve the business objects, tables, views, columns, keys, and foreign keys from PeopleSoft. The database hosting the PeopleSoft tables is Microsoft SQL Server.

Installation and Configuration

Make sure you have read the information in this section before you start working with the PeopleSoft technology:

- [System Requirements and Certifications](#)
- [Technology Specific Requirements](#)
- [Connectivity Requirements](#)

System Requirements and Certifications

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 list of supported platforms and versions is available on Oracle Technical Network (OTN):

<http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html>.

Technology Specific Requirements

There are no technology-specific requirements for using PeopleSoft data in Oracle Data Integrator.

Connectivity Requirements

Oracle Data Integrator connects the database hosting the PeopleSoft data using JDBC connectivity. For detailed information on JDBC connectivity with Oracle and Microsoft SQL Server databases, see the Oracle Database Connectivity Requirements and Microsoft SQL Server Connectivity Requirements in the *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator*.

Setting up the Topology

This step consists in declaring in Oracle Data Integrator the data server, physical schema and logical schema that will be used to store the PeopleSoft data.

Create a Data Server

The PeopleSoft tables can be stored in an Oracle schema or a Microsoft SQL Server database.

Create a data server either for the Oracle technology or for the Microsoft SQL Server technology. See [Creating an Oracle Data Server](#) or [Creating a Microsoft SQL Server Data Server](#) in the *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator*.

This data server represents the database instance that stores the PeopleSoft data.

Create a Physical Schema

Create a physical schema under the data server that you have created in [Create a Data Server](#). Use the standard procedure, as described in [Creating a Physical Schema](#) in *Administering Oracle Data Integrator*.

This schema is the Oracle schema or the Microsoft SQL Server database that contains the PeopleSoft tables you want to reverse-engineer.

Note:

The Oracle schema or the Microsoft SQL Server database storing the PeopleSoft tables should not be defined as a work schema in the physical schema definition. Moreover, this schema or database must not be used as staging area for a mapping.

Create for this physical schema a logical schema using the standard procedure, as described in [Creating a Logical Schema](#) in *Administering Oracle Data Integrator* and associate it in a given context.

Setting up the Project

Setting up a project using PeopleSoft features follows the standard procedure. See [Creating an Integration Project](#) of the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

Import the following KMs into your Oracle Data Integrator project:

- RKM PeopleSoft ORACLE
- RKM PeopleSoft MSSQL

In addition to these specific PeopleSoft KMs, import the standard LKMs for the Oracle or Microsoft SQL Server technologies. See [Oracle Database Knowledge Modules and Microsoft SQL Server Knowledge Modules](#) in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for a list of available KMs.

Creating and Reverse-Engineering a Model

This section contains the following topics:

- [Create a Model](#)
- [Reverse-Engineer PeopleSoft Tables](#)

Create a Model

Create a Model based on the Oracle or Microsoft SQL technology and on the logical schema created when setting up the topology using the standard procedure, as described in *Creating a Model of the Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

Reverse-Engineer PeopleSoft Tables

The PeopleSoft RKMs are able to reverse-engineer PeopleSoft data structures at data level, enriching them with information retrieved from the PeopleSoft dictionary.

Data extraction is performed directly on the PeopleSoft Business Objects tables. This access method is read-only.

The reverse-engineering process returns the following information:

- Business Objects as sub-models
- Business Objects tables as datastores with their associated columns as attributes and constraints
- Comments attached to the reversed tables and columns

To perform a Customized Reverse-Engineering of PeopleSoft tables with the PeopleSoft RKMs, use the usual procedure, as described in *Reverse-engineering a Model of the Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*. This section details only the fields specific to PeopleSoft tables:

1. In the Reverse tab of the Model, select the RKM PeopleSoft ORACLE. or RKM PeopleSoft MSSQL.
2. Set the BUSINESS OBJECT RKM option as follows:

Enter the Business Object code, for example CCM, DBI, or DPO.

The Business Object code corresponds to the PeopleSoft *Object Owner ID*. The different Object Owner IDs are listed in the PeopleSoft view: `EO_BCONNRID_VW`. This field is used as a mask to filter the Business Objects to be reverse-engineered. This field must not be empty and must contain at least the percentage symbol (%).

3. Specify the reverse-engineering mask in the Mask field in order to select the tables to reverse. The Mask field, in the Reverse tab, filters reverse-engineered objects based on their name. The Mask field must not be empty and must contain at least the percentage symbol (%).

The reverse-engineering process returns the applications and tables as sub-models and datastores. You can use these PeopleSoft datastores as a source in a mapping.

Designing a Mapping

You can use PeopleSoft data tables as a source of a mapping.

The KM choice for a mapping determines the abilities and performance of this mapping. The recommendations in this section help in the selection of the KM for different situations concerning loading PeopleSoft data.

Loading Data from PeopleSoft

PeopleSoft data tables can be used as a source of a mapping to extract data from the PeopleSoft database and integrate them into another system (Data warehouse, other database...).

Using PeopleSoft as a source in these conditions is identical to using an Oracle or Microsoft SQL Server datastore as a source in a mapping. The generic SQL, Oracle Database, or Microsoft SQL Server KMs can be used for this purpose. See the following chapters in the *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for more information:

- Oracle Database
- Generic SQL
- Microsoft SQL Server

6

Oracle Siebel CRM

Oracle Data Integrator provides specific Knowledge Modules (KMs) for handling Siebel CRM data. You can obtain information on how to work with these KMs in this section. This chapter includes the following sections:

- [Introduction](#)
- [Installation and Configuration](#)
- [Setting up the Topology](#)
- [Setting up the Project](#)
- [Creating and Reverse-Engineering a Model](#)
- [Designing a Mapping](#)

Introduction

Oracle Data Integrator can integrate Oracle Siebel CRM applications. It supports reverse-engineering of Siebel metadata as well as data extraction on Siebel Business Components tables and integration through EIM tables.

Concepts

The Oracle Data Integrator Siebel Knowledge Modules (KMs) use mature integration methods for Siebel, in order to:

- Reverse-Engineer Siebel data structures (Business Components and Business Objects)
- Reverse-Engineer EIM (Enterprise Integration Manager) tables
- Read data from Siebel using data-level integration
- Read and write Siebel data using the EIM tables

These features are supported for Oracle and Microsoft SQL Server (MSSQL) databases used as back-end databases for Siebel. In this chapter, <database> refers to any of these technologies.

Overview of Extracting Data from Siebel

Oracle Data Integrator provides two ways to extract data from Siebel:

- [Data-level integration](#)
- [Integration through EIM tables](#)

Data-level integration

Data extraction is performed directly on the Siebel Business Components tables. You can use a Siebel data model as a source of a mapping by extracting data from the

Siebel Database and integrate them into another system. Using Siebel as a source in these conditions is the same as using a regular table as a source in a mapping. This integration method is read-only.

- *Reverse-engineering*: To reverse-engineer Siebel Business Components, use the RKM Siebel <database> (<database> is the name of the database hosting the Siebel tables). This RKM allows for reverse-engineering of the Siebel data structures, enriching them with information retrieved from the Siebel dictionary.
- *Extracting*: You have access to a range of knowledge modules to extract Siebel data from Siebel. The Generic SQL, Oracle Database or Microsoft SQL Server KMs can be used for this purpose.

Integration through EIM tables

The EIM tables are used to extract data from Siebel and load data to Siebel. EIM tables act as a staging area between the Siebel application and the other applications (another Siebel can be one of these applications). This method supports read and write.

- *Reverse-engineering*: To reverse-engineer Siebel EIM tables, use the RKM Siebel EIM <database>. This RKM allows for reverse-engineering of the Siebel EIM tables, enriching them with information retrieved from the Siebel dictionary.
- *Extracting*: Data extraction is performed on the EIM tables after executing automatically an export script to load these EIM tables from the Siebel application tables. To extract data from a Siebel EIM table and load it to any SQL staging area, use the LKM Siebel to SQL (EIM).

This LKM first generates the appropriate EIM Configuration file (.ifb – Interface Builder File) and runs the export process using the Server Manager. Then, it extracts selected data from the EIM Siebel table into the staging area.

- *Integrating*: Data integration is performed to the EIM tables and then an import script is generated and executed to import data from the EIM tables into the Siebel application tables. To perform such integration, from the staging area, use the IKM SQL to Siebel Append (EIM).

This IKM first loads data from the staging area to the EIM table. Then, it generates the EIM configuration file (.ifb) and runs the import process using the server Manager.

Note:

Only EIM tables should be used to write into Siebel. Writing directly into the Siebel physical tables is not recommended.

Knowledge Modules

Oracle Data Integrator provides the Knowledge Modules listed in [Table 6-1](#) for handling Siebel CRM data. These specific Knowledge Modules for Oracle Siebel CRM provide integration and connectivity between Oracle Data Integrator and the Oracle Siebel CRM platform.

Table 6-1 Siebel CRM Knowledge Modules

Knowledge Module	Description
IKM SQL to Siebel Append (EIM)	Integrates data into a Siebel EIM (Enterprise Integration Manager) table from any ANSI-SQL92 compliant staging area, then generates the appropriate EIM configuration files (.ifb) and runs the import process using the Siebel Server Manager. The target table is populated in truncate/insert mode.
LKM Siebel to SQL (EIM)	Loads data from a Siebel EIM (Enterprise Integration Manager) table to any ANSI-SQL92 compliant staging area. This module uses the run-time Agent to extract data from EIM (Enterprise Integration Manager) table to the staging area. It can generate the appropriate EIM configuration files (.ifb) and run the export process using the Siebel Server Manager.
RKM Siebel Oracle	Reverse-engineering knowledge module for Siebel. Business Objects are reversed as sub-models, Business Components are reversed as datastores with their columns as attributes and their constraints (Primary and Foreign Keys).
RKM Siebel EIM Oracle	Reverse-engineering knowledge module for Siebel EIM (Enterprise Integration Manager) tables. Siebel projects are reversed as sub-models, EIM tables are reversed as datastores with their columns as attributes and Primary Keys.
RKM Siebel MSSQL	This RKM provides the same features as the RKM Siebel Oracle for Siebel installed on top of a Microsoft SQL Server database.
RKM Siebel EIM MSSQL	This RKM provides the same features as the RKM Siebel EIM Oracle for Siebel installed on top of a Microsoft SQL Server database.

Installation and Configuration

Make sure you have read the information in this section before you start working with the Siebel CRM data:

- [System Requirements and Certifications](#)
- [Technology Specific Requirements](#)
- [Connectivity Requirements](#)

System Requirements and Certifications

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 list of supported platforms and versions is available on Oracle Technical Network (OTN):

<http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html>.

Technology Specific Requirements

In order to use the Siebel EIM KMS, it is required that the Srvrmgr Siebel utility is installed on the machine hosting the run-time agent.

Connectivity Requirements

Oracle Data Integrator connects the database hosting the Siebel CRM data using JDBC connectivity. For detailed information on JDBC connectivity with Oracle and Microsoft SQL Server databases, see the Oracle Database Connectivity Requirements and Microsoft SQL Server Connectivity Requirements in the *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator*.

Setting up the Topology

This step consists in declaring in Oracle Data Integrator the data server, physical schema and logical schema that will be used to store the Siebel CRM data.

Create a Data Server

The Siebel CRM tables can be stored in an Oracle schema or a Microsoft SQL Server database.

Create a data server either for the Oracle technology or for the Microsoft SQL Server technology. See *Creating an Oracle Data Server* or *Creating a Microsoft SQL Server Data Server* in the *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator*.

This data server represents the database instance that stores the Siebel CRM data.

Create a Physical Schema

Create a physical schema under the data server that you have created in [Create a Data Server](#) . Use the standard procedure, as described in *Creating a Physical Schema of the Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

This schema must point to the Oracle schema or Microsoft SQL Server database that contains the Siebel tables or EIM data structures you want to reverse-engineer.

 **Note:**

The Oracle schema or the Microsoft SQL Server database storing the Siebel CRM tables should not be defined as a work schema in the physical schema definition. Moreover, this schema or database must not be used as staging area for a mapping.

Create for this physical schema a logical schema using the standard procedure, as described in *Creating a Logical Schema of the Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator* and associate it in a given context.

Setting up the Project

Setting up a project using Siebel CRM features follows the standard procedure. See *Creating an Integration Project* of the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

Import the following KMs into your Oracle Data Integrator project:

- RKM Siebel <database> or RKM Siebel EIM <database>
- LKM Siebel to SQL (EIM)
- IKM SQL to Siebel Append (EIM)

In addition to these specific Siebel CRM KMs, import the standard LKMs for the Oracle or Microsoft SQL Server technologies. See *Oracle Database Knowledge Modules and Microsoft SQL Server Knowledge Modules* in the *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for a list of available KMs.

Creating and Reverse-Engineering a Model

This section contains the following topics:

- [Create a Model](#)
- [Reverse-Engineer Siebel CRM Tables](#)

Create a Model

Create a Model based on the Oracle or Microsoft SQL Server technology and on the logical schema created when setting up the topology using the standard procedure, as described in *Creating a Model* of the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

Reverse-Engineer Siebel CRM Tables

The Siebel RKMs are able to reverse-engineer Siebel Business Components or Siebel EIM tables, enriching them with information retrieved from the Siebel dictionary.

The reverse-engineering process using the RKM Siebel <database> returns:

- The installed Business Objects as sub-models
- The Business Components as datastores with their columns as attributes and their constraints (Primary and Foreign Keys)
- Comments on the reversed tables and columns

The reverse-engineering process using the RKM Siebel <database> EIM returns:

- Projects as sub-models
- EIM tables as datastores with their columns as attributes and their constraints (Primary and Foreign Keys).

To perform a Customized Reverse-Engineering of Siebel CRM tables with the Siebel CRM RKMs, use the usual procedure, as described in *Reverse-engineering a Model* of

the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*. This section details only the fields specific to Siebel CRM tables:

1. In the Reverse tab of the Model, select depending on the integration method you want use, the RKM Siebel <database> or RKM Siebel EIM <database>.
2. The RKM options vary depending on the RKM selected in step 1. When using the:
 - a. RKM Siebel to Oracle or the RKM Siebel MSSQL, specify the mask to filter the Business Objects to reverse-engineer in the Business Object option.
For example: Account, Acc%, Customer, Employee, %empl%.
The percent sign (%) returns all Business Objects.
 - b. RKM Siebel EIM Oracle or the RKM Siebel EIM MSSQL, set the options as shown in [Table 6-2](#).

Table 6-2 KM options for Siebel EIM Knowledge Modules

Option	Description
USE_PROJECT	Set this option to YES to reverse-engineer projects as sub-models in Oracle Data Integrator.
REPOSITORY	Specify the Siebel Repository name. Default is Siebel Repository.
PROJECT_NAME	Specify the mask to filter the Siebel projects to reverse-engineer. For example: EIM Accounts and Quotes, EIM Activity, EIM A%. The percent sign (%) returns all Siebel projects

3. Specify the reverse-engineering mask in the Mask field in order to select the tables to reverse. The Mask field, in the Reverse tab, filters reverse-engineered objects based on their name. The Mask field must not be empty and must contain at least the percentage symbol (%).

The reverse-engineering process returns the applications and tables as sub-models and datastores. You can use these Siebel CRM datastores as a source in a mapping.

Designing a Mapping

You can use Siebel Business Components tables as sources in mappings. EIM tables can be used as sources or targets.

The KM choice for a mapping determines the abilities and performance of this mapping. The recommendations in this section help in the selection of the KM for different situations concerning loading and integrating Siebel CRM data.

Loading Data from Siebel

After performing a reverse-engineering using the RKM Siebel <database>, you can use Siebel data tables as sources in mappings.

Using Siebel CRM as a source in these conditions is identical to using an Oracle or Microsoft SQL Server datastore as a source in a mapping. The generic SQL, Oracle Database, or Microsoft SQL Server KMs can be used for this purpose. See the

following chapters in the *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for more information:

- Oracle Database
- Generic SQL
- Microsoft SQL Server

Loading Data from Siebel through EIM tables

To extract data from Siebel through the EIM tables, create a mapping with EIM tables as a source. Select the LKM Siebel to SQL (EIM) and set the KM options as follows:

- **IFB_PATH:** Specify the path where you want to create the EIM configuration file (.ifb).
- **SRVRMGR_PATH:** Specify the location of the Siebel srvmgr binary. This parameter is mandatory.
- **SIEBEL_GATEWAY:** Specify the network address of the Gateway Server machine.
- **SIEBEL_ENTERPRISE:** Indicate the name of the Enterprise Server.
- **SIEBEL_SERVER:** Indicate the name of the Siebel Server.
- **SERVER_USER:** Indicate the user name of the Server administrator.
- **SERVER_USER_PWD:** Indicate the Server administrator password.

The LKM Siebel to SQL (EIM) automatically performs the following operations:

1. Generate an EIM Configuration File, describing the export process to the EIM tables.
2. Run the EIM process using for example the Siebel srvmgr command line.
3. Extract, transform and load data from the EIM tables to the other application.

Integrating Data in Siebel through EIM tables

To insert data into Siebel through the EIM tables, create a mapping with EIM tables as target. Select the IKM SQL to Siebel Append (EIM) and set the KM options as follows:

- **IFB_PATH:** Specify the path where you want to create the EIM configuration file (.ifb).
- **SRVRMGR_PATH:** Specify the location of the Siebel srvmgr binary. This parameter is mandatory.
- **SIEBEL_GATEWAY:** Specify the network address of the Gateway Server machine.
- **SIEBEL_ENTERPRISE:** Indicate the name of the Enterprise Server.
- **SIEBEL_SERVER:** Indicate the name of the Siebel Server.
- **SERVER_USER:** Indicate the user name of the Server administrator.
- **SERVER_USER_PWD:** Indicate the Server administrator password.

The IKM SQL to Siebel Append (EIM) automatically performs the following operations:

1. Load the appropriate EIM tables.

2. Generate an EIM Configuration File, describing the import process from the EIM tables.
3. Run the EIM process using for instance the Siebel `srvmgr` command line.

7

SAP ABAP BW

Oracle Data Integrator provides specific Knowledge Modules (KMs) for handling SAP BW data. You can obtain information on how to work with these KMs in this section. This chapter includes the following sections:

- [Introduction](#)
- [Installation and Configuration](#)
- [Defining the Topology](#)
- [Setting up the Project](#)
- [Creating and Reverse-Engineering a Model](#)
- [Designing a Mapping](#)
- [Considerations for SAP BW Integration](#)

Introduction

The SAP BW Knowledge Modules let Oracle Data Integrator connect to SAP-BW system using SAP Java Connector (SAP JCo) libraries. These adapters allow mass data extraction from SAP-BW systems.

If this is the first time you are using the SAP BW adapter, it is recommended to review *Getting Started with SAP ABAP BW Adapter for Oracle Data Integrator*.

It contains the complete pre-requisites list and step-by-step instructions including SAP connection testing.

Concepts

The SAP BW Knowledge Modules for Oracle Data Integrator use mature integration methods for SAP-BW system, in order to:

- Reverse-Engineer SAP BW metadata
- Extract and load data from SAP BW system (source) to an Oracle or non-Oracle Staging Area

The reverse-engineering process returns the following SAP BW objects inside an ODI model:

- Each ODS/DSO object is represented as an ODI datastore.
- Each InfoObject will be represented in ODI as a submodel containing up to three datastores:
 - InfoObjects having master data have a master data datastore containing all InfoObject attributes
 - InfoObjects having attached text data have a text datastore containing all text related attributes

- InfoObjects having hierarchies defined have a hierarchy datastore containing all hierarchy related attributes
- Each InfoCube will be represented as a single ODI datastore. This datastore includes attributes for all characteristics of all dimensions as well as for all key figures.
- Each OpenHubDestination is represented as an ODI datastore.

Knowledge Modules

Oracle Data Integrator provides the Knowledge Modules listed in [Table 7-1](#) for handling SAP BW data.

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 BW data from SAP BW system.
- Loads this data into Oracle or non-Oracle Staging Area.
- Reverse-engineers SAP Metadata and proposes a tree browser to select only the required Metadata.
- Uses flexfields to map the SAP BW data targets types (InfoCube, InfoObject, ODS/DSO, OpenHub and Text Table) and their attributes.

Table 7-1 SAP BW Knowledge Modules

Knowledge Module	Description
LKM SAP BW to Oracle (SQLLDR)	Extracts data from SAP BW system into a flat file and then loads it into Oracle Staging Area using the SQL*LOADER command line utility.
RKM SAP ERP Connection Test	This RKM is used for testing the SAP connection from Oracle Data Integrator. See Additional Information for SAP ABAP ERP Adapter for more information.
RKM SAP BW	Reverse-engineering Knowledge Module to retrieve SAP specific metadata for InfoCubes, InfoObjects (including Texts and Hierarchies), ODS/DSO and OpenHubDestinations.
LKM SAP BW to SQL	Extracts data from SAP BW into a flat file and then loads it into a Staging Area using a JDBC connection.

Overview of the SAP BW Integration Process

The RKM SAP BW enables Oracle Data Integrator (ODI) to connect to SAP BW system using SAP JCo libraries and perform a customized reverse-engineering of SAP BW metadata.

The LKM SAP BW to Oracle (SQLLDR) and LKM SAP BW to SQL are in charge of extracting and loading data from SAP BW system (source) to an Oracle or non-Oracle Staging Area.

 **Note:**

Access to SAP BW is made through ABAP. As a consequence, the technology used for connecting is SAP ABAP, and the topology elements, as well as the model will be based on the SAP ABAP technology. There is no SAP BW technology in ODI, but SAP BW-specific KMs based on the SAP ABAP technology.

Reverse-Engineering Process

Reverse-engineering uses the RKM SAP BW.

This knowledge module extracts the list of SAP BW data objects and optionally displays this list in a Metadata Browser graphical interface. The user selects from this list the SAP BW objects to reverse-engineer.

In the reverse-engineering process, data targets, primary keys, foreign keys and index are reverse-engineered into an Oracle Data Integrator model.

Integration Process

Data integration from SAP is managed by the LKM SAP BW to Oracle (SQLLDR) and the LKM SAP BW to SQL.

The LKM SAP BW to Oracle (SQLLDR) is used for mappings sourcing from SAP via ABAP and having a Staging Area located in an Oracle Database and the LKM SAP BW to SQL is used for non-Oracle staging areas.

The KM first generates optimized ABAP code corresponding to the extraction process required for a given mapping. This code includes filters and joins that can be processed directly in the source SAP BW server. This ABAP program is automatically uploaded and is executed using the OdiSAPAbapExecute tool to generate an extraction file in SAP.

The KM then transfers this extraction file either to a pre-configured FTP server or to a shared directory. This file is then either downloaded from this server using FTP, SFTP, SCP or copied to the machine where the ODI Agent is located, and is finally loaded either using SQL*Loader or using a JDBC connection to the staging area. The agent can also directly read the extraction file on the FTP server's disk. See [File Transfer Configurations](#) for more information.

The rest of the integration process (data integrity check and integration) is managed with other Oracle Data Integration KMs.

Installation and Configuration

Make sure you have read the information in this section before you start working with the SAP BW data:

- [System Requirements and Certifications](#)
- [Technology Specific Requirements](#)
- [Connectivity Requirements](#)

System Requirements and Certifications

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 list of supported platforms and versions is available on Oracle Technical Network (OTN):

<http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html>.

Technology Specific Requirements

Some of the Knowledge Modules for SAP BW use specific features of SAP-BW system and Oracle database. This section lists the requirements related to these features.

- A JCo version compatible with adapter must be used. The list of supported JCo versions is available from the Oracle Technology Network (OTN). See [System Requirements and Certifications](#) for more information.
- A JVM version compatible with both Oracle Data Integrator and JCo must be used.
- The adapter supports two transfer modes for transferring data from SAP system to the ODI agent: data transfer using a Shared Directory and data transfer through FTP. For details and restrictions, see [File Transfer Configurations](#).

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, folders located on the SAP server should not be shared. You should instead share a folder located on the ODI agent machine with the SAP system, or use a third machine as the shared file server.

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: SQL*Loader is required on the machine running the agent when executing mappings using LKM SAP BW to Oracle (SQLLDR). SQL*Loader is used for loading data extracted from SAP to the Oracle staging area.

Connectivity Requirements

Oracle Data Integrator connects to the SAP BW system hosting the SAP BW data using JCo. It also uses a FTP Server or a shared directory to host the data extracted from the SAP system.

This section describes the required connection information:

- [Installing and Configuring JCo](#)
- [Installing ODI SAP Components into SAP System](#)
- [Requesting FTP Server Access](#)
- [Validating the SAP Environment Setup](#)
- [Gathering SAP Connection Information](#)
- [Gathering FTP Connection Information](#)
- [Gathering Shared Directory Information](#)
- [Adding the Open Tool](#)

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 Technical Network:

<http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html>

 **Note:**

- A minimum version of JCo 3.0.2 is required
 - 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 provided in the JCo documentation (`<sapjco-install-path>/javadoc/installation.html`) for your platform.
 4. Copy the required files (`sapjco3.jar` and `sapjco3.dll/.so`) into the `<ODI_HOME>/odi/sdk/lib` directory (ODI Studio, ODI Standalone Agent) and into the `<WLS_DOMAIN>/lib` directory (JEE Agent).
 5. Restart the ODI Components using SAP (ODI Studio, Standalone Agent)
 6. Check the JCo installation.

 **Note:**

For SAP Secure Network Connections, ensure the following:

- the environment variable `SECUDIR` points to the directory containing any certificate files.
- the environment variable `SNC_LIB` points to the directory containing `sapcrypto.dll/libsapcrypto.so` for Linux.

Please contact your SAP Administrator for more details on certificates and Crypto Libraries.

Installing ODI SAP Components into SAP System

The ODI SAP adapter communicates with the SAP System using a few ODI SAP RFCs. These RFCs are installed by your SAP Basis team using SAP Transport requests. Please contact your SAP administrators for installing the ODI SAP Components and assigning the required SAP user authorizations following the instructions given in [Installing ODI SAP Components](#).



Requesting FTP Server Access

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

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

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

To request FTP server access:

1. Login to SAP server.
2. Execute transaction SE16.
3. Enter `SAPFTP_SERVERS` in the **Table Name** field and click  icon to create entries.
4. Enter IP address of the FTP server in the **FTP_SERVER_NAME** field.
5. Enter port of the FTP server in the **FTP_Server_Port** field.
6. Click  icon to save.

Validating the SAP Environment Setup

[Validating the ODI SAP Setup](#) contains instructions for some basic validation of the SAP environment for the use with the ODI SAP Adapter. Please ask your SAP Basis team to run all validations and provide back the validation results like screen shots and confirmations.

Gathering SAP Connection Information

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

- for SAP Group Logons:
 - SAP System ID: The three-character, unique identifier of a SAP system in a landscape.
 - Message Server: IP address/ host name of SAP Message Server.
 - Message Server Port: port number or service name of SAP Message Server.
 - Group/Server: Name of SAP Logon Group.
- for SAP Server Logons:
 - SAP BW System IP Address or Hostname: IP address/ host name of the host on which SAP is running.
 - 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 BW System IP Address or Hostname: IP address/ host name of the host on which SAP is running.

- SAP User: SAP User is the unique user name given to a user for logging on the SAP System.
- SAP Password: Case-sensitive password used by the user to log in.
- SAP Language: Code of the language used when logging in For example: EN for English, DE for German.
- SAP Client Number: The three-digit number assigned to the self-contained unit which is called Client in SAP. A Client can be a training, development, testing or production client or represent different divisions in a large company.
- SAP System Number: The two-digit number assigned to a SAP instance which is also called Web Application Server or WAS.
- SAP System ID: The three-character, unique identifier of a SAP system in a landscape.
- SAP SNC Connection Properties (optional) SAP Router String (optional): SAP is enhancing security through SNC and SAP router. It is used when these securities are implemented.
- SAP Transport Layer Name: This string uniquely identifies a transport layer in a SAP landscape. It allows ODI to create transport requests for later deployment in SAP. Even though there is a default value here, this transport layer name *must* be provided by your SAP Basis team. Not doing so may result in significant delays during installation.
- SAP Temporary Directory (`SAP_TMP_DIR`): This parameter is used to define the custom work directory on the SAP system. It applies only in case of FTP transfer mode. This path is used for temporary files during extraction. The default value (FlexField value is left empty) is to use the `DIR_HOME` path defined in the SAP profile.

Specify a directory on the SAP application server to be used for temporary files. The path must end on slash or backslash depending on the OS the SAP application server runs on. The ODI SAP user must have read and write privileges on this directory.

 **Caution:**

The default value (empty FF value) must be avoided for any critical SAP system, as an excessive temp file during extraction could fill up the respective file system and can cause serious issues on the SAP system. Even for SAP development systems it is strongly recommended to override the default value.

- SAP BW Version: 3.5 or 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, `EE8IS08859P2` 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.

Gathering FTP Connection Information

The SAP BW system will push data to a server using the FTP protocol. Collect the following information from your system administrator:

- FTP server name or IP address
- FTP login ID
- FTP login password
- Directory path for storing temporary data files

Validate that the FTP server is accessible both from SAP System and from ODI agent machine.

Gathering Shared Directory Information

Gathering Shared Directory information only applies, if you plan to transfer data through a shared directory. The SAP system will push data to a shared folder. For later setup, gather the following information from your system administrator:

- (UNC) path name of the shared folder

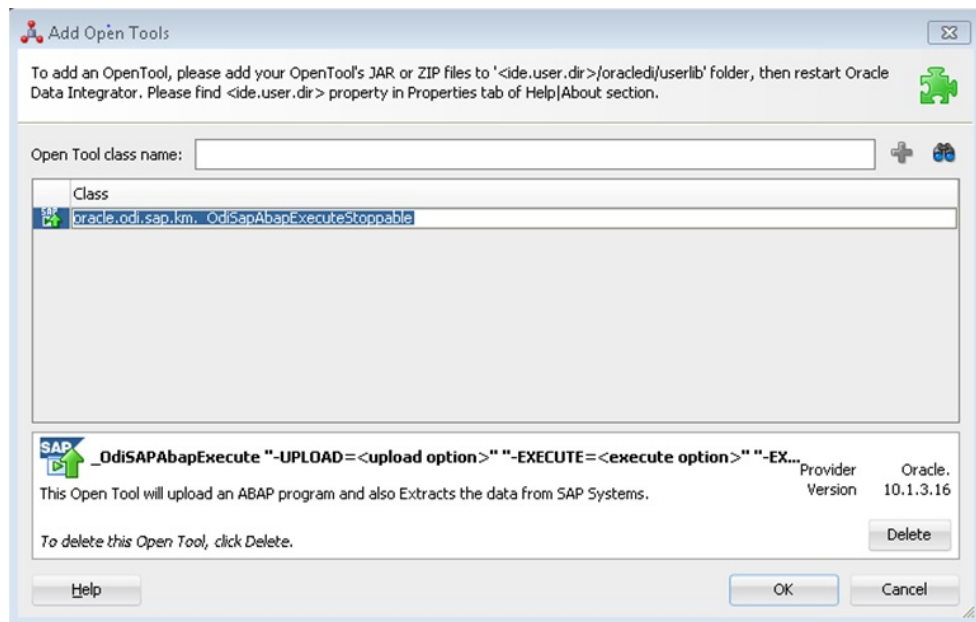
Validate that the shared folder is accessible from both the SAP System and the ODI agent machine and does not require any interactive authentication to be accessed.

Please note that the shared folder must be accessible from the SAP system using the `<sid>adm` user and from the operating system user that starts the ODI agent.

Adding the Open Tool

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

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



6. Click **OK**.

Defining the Topology

You must define the two data servers used for SAP integration. The SAP ABAP Data Server and the FTP data server.

1. [Create the File Data Server](#)
2. [Create the SAP ABAP Data Server](#)

Create the File Data Server

This data server corresponds to the FTP server or File Server into which the extraction file will be pushed from SAP and picked up for SQL*Loader / JDBC driver.

Create a File Data Server

Create a File data server as described in *Creating a File Data Server of Connectivity and Knowledge Modules Guide for Oracle Data Integrator*. This section describes the parameters specific to SAP BW.

Depending on the chosen data transfer mode, this data server must point either to:

- An existing FTP server into which the extraction file will be pushed from SAP and picked up for loading or
- The shared folder into which the SAP system will write the extraction file and from which SQL*Loader/ ODI Flat File Driver will pick it up. This schema represents the folder in the FTP host where the extraction file will be pushed.

Note that the parameters for the data server depend on the data transfer mode.

- When transferring data through FTP, set the parameters 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
- When transferring data through a Shared Directory, set the parameters as follows:
 - Host (Data Server): n/a
 - User: n/a
 - Password: n/a
- 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, refer details on ENCODING parameter in Creating a File Data Server of *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*. The encoding chosen on this URL must match the code page used by the SAP Application Server.

See [File Transfer Configurations](#) for more information.

Create the File Schema

In this File data server create a Physical Schema as described in Creating a File Physical Schema of the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*.

This schema represents the folder that is located either in the FTP host or the File Server. Note that this is the folder into which the extraction file will be pushed. Depending on the data transfer mode, specify the Data and Work Schemas as follows:

- For the FTP file transfer:
 - *Directory (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.
 - *Directory (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 FTP. 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. The Work Schema is a local directory location. As a consequence, slashes or backslashes should be used according to the operating system. This path must end with a slash or backslash.

Path names given on Data and Work schemas are not necessarily the same: the FTP server may provide access to a FTP directory named /sapfiles - the value for Directory (Schema) - while the files are accessed locally in c:\inetpub\ftproot\sapfiles - the value for Directory (Work Schema).

- For the Shared Directory transfer:
 - *Directory (Schema)*: Path (UNC) of the shared folder to write and read extraction files. SAP System writes the extraction files into this folder. It is also

used by a remote agent to copy the extraction files to the ODI agent machine. Note that this path must use slashes or backslashes according to the operating system of the SAP Application Server and must end with a slash or backslash character.

- *Directory (Work Schema)*: Local path on the server's machine hosting the shared folder. This path is used by an agent installed on this machine to access the extraction files without passing through the shared folder. This access method is used if the `FTP_TRANSFER_METHOD` parameter of the LKM SAP BW to Oracle (SQLDR)/ LKM SAP BW to SQL is set to `FSMOUNT_DIRECT`. The Work Schema is a local directory location. As a consequence, slashes or backslashes should be used according to the operating system. This path must end with a slash or backslash.

See [File Transfer Configurations](#) for more information.

Create a File Logical Schema called `File Server for SAP ABAP`, and map it to the Physical Schema. The name of this Logical Schema name is predefined and must be `File Server for SAP ABAP`.

Create the SAP ABAP Data Server


This SAP ABAP data server corresponds to the SAP server from which data will be extracted.

Create the SAP ABAP Data Server

To configure a SAP ABAP data server:

1. Create a data server for the SAP ABAP technology using the standard procedure, as described in *Creating a Data Server of the Developing Integration Projects with Oracle Data Integrator*. This data server uses the SAP connection information.
2. Set the connection parameters:
 - Name: The name of the data server as it will appear in ODI. For example, `SAP_BW`.
 - User: SAP BW User, as provided by the SAP Administrator.
 - Password: This user's SAP BW Password. This password is case-sensitive.
3. Set Logon type specific connection parameters:
 - either for SAP Group Logons (on FlexField tab):
 - FlexField SAP System ID: The three-character, unique identifier of a SAP system in a landscape.
 - FlexField SAP Message Server Host: IP address/ host name of SAP Message Server.
 - FlexField SAP Message Server Port: port number or service name of SAP Message Server.
 - FlexField SAP Group: Name of SAP Logon Group.
 - or for SAP Server Logons:
 - Host (Data Server): SAP BW System IP Address or host name.
 - FlexField 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.

- FlexField SAP System Number: The two-digit number assigned to a SAP instance which is also called Web Application Server or WAS.
 - FlexField SAP System ID: The three-character, unique identifier of a SAP system in a landscape.
4. Set SAP Secure Network Settings (optional):
 - FlexField "for SNC: Level of Security": 0 for "No SNC", 1 for "Authentication", 2 for "Integrity", 3 for "Encryption", 9 for "Maximum Security Settings Available"
 - FlexField "for SNC: Client Name": SNC Client Name
 - FlexField "for SNC: Partner Name": SNC Partner Name
 5. Set the flexfields 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 Router String: Router String. This parameter is optional and can be left empty.
 - SAP Character Set: The character set is only required if your SAP system is not a UNICODE system. For a complete list of character sets, see Locale Data in the *Oracle Database Globalization Support Guide*. For example, `EE8ISO8859P2` for Croatian Data. For UNICODE systems, use `UTF8`.
 - SAP Temporary Directory: In case of FTP based file transfer (compared to shared folder transfer) data is extracted to the default temp directory of the SAP server. If needed, a different folder can be used instead.
 - SAP Temporary Directory Name (Logical) (optional): While **SAP Temporary Directory** specifies a *physical* SAP directory, the FlexField **SAP Temporary Directory (logical)** can be used to specify a *logical* SAP directory. During the ABAP execution the logical path is translated into the physical path by the SAP system.
-  **Note:**

Please note the following:

 - Only either **SAP Temporary Directory Name** or **SAP Temporary Directory Name (Logical)** can be specified.
 - The physical path defined in the SAP system must end on [back]slash.
- SAP Extract Timeout: Maximum number of seconds ODI will wait for a SAP background job before it times out.
 - 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: not used in SAP BW Connector.

- SAP Allow ABAP Upload: set to 0 for any non-development system. See [Controlling ABAP Uploading / ABAP code in production](#) for more details.
- SAP Allow ABAP Execute: set to 1, if ABAP code can be executed on this SAP System. See [Controlling ABAP Uploading / ABAP code in production](#) for more details.

 **Note:**

The Test button for validating the SAP Connection and the FTP Connection definition is not supported.

- SAP Transport Layer Name: This transport layer is used by ODI to create any new transport requests. The default transport layer name is SAP. If the SAP system uses a different transport layer, this FlexField must be updated accordingly. Otherwise, any TR creation will fail.

Except for Data Server Name, all the parameters that you provide while defining the SAP Data Server should be provided by the SAP Administrators. See [Gathering SAP Connection Information](#) for more information about these parameters.

Create the SAP ABAP Schema

To configure a SAP ABAP schema:

1. Create a Physical Schema under the SAP ABAP data server as described in Creating a Physical Schema in *Administering Oracle Data Integrator*. This schema does not require any specific configuration. Only one physical schema is required under a SAP ABAP data server.
2. Create a Logical Schema for this Physical Schema as described in Creating a Logical Schema in *Administering Oracle Data Integrator* in the appropriate context.

Setting up the Project

Setting up a project using SAP BW features follows the standard procedure. See Creating an Integration Project of the *Developing Integration Projects with Oracle Data Integrator*.

Import the following KMs into your Oracle Data Integrator project:

- RKM SAP BW
- LKM SAP BW to Oracle (SQLLDR)
- LKM SAP BW to SQL

In addition to these specific SAP BW KMs, import the standard Oracle LKMs, IKMs, and CKMs to perform data extraction and data quality checks with an Oracle database. See Oracle Database in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for a list of available KMs.

Creating and Reverse-Engineering a Model

This section contains the following topics:

- [Creating a SAP BW Model](#)
- [Reverse-Engineering a SAP BW Model](#)
- [Reverse-Engineering a PSA Table in SAP BW Model](#)

Creating a SAP BW Model

Create an SAP BW Model based on the SAP ABAP technology and on the SAP ABAP logical schema using the standard procedure, as described in *Creating a Model of the Developing Integration Projects with Oracle Data Integrator*.

Reverse-Engineering a SAP BW Model

To perform a Customized Reverse-Engineering with the RKM SAP BW, use the usual procedure, as described in *Reverse-engineering a Model of the Developing Integration Projects with Oracle Data Integrator*. This section details only the fields specific to SAP BW:

1. In the Reverse Engineer tab of the SAP BW Model, select the RKM SAP BW.
2. For the RKM SAP BW, set the USE_GUI KM option to `true`.
3. Save the model.
4. Click **Reverse-Engineer** in the Model Editor toolbar.
5. The Tree Metadata Browser appears after the session is started. Select the data store object(s) to reverse.
6. Click **Reverse-Engineer** in the Tree Metadata Browser window.

The reverse-engineering process returns the selected data store objects as `datastores`.

Note:

If the reverse-engineering is executed on a run-time agent, the USE_GUI option should be set to `false`. This option should be used only when the customized reverse-engineering is started using the agent built-in the Studio.

Reverse-Engineering a PSA Table in SAP BW Model

PSA tables are used to store data of a data source in BW system. Generally when we run load (Extraction) of a data-source in BW system, it gets all active metadata information from partner ERP system, and replicate this in BW system.

For example if we have 10 fields in ERP system and now we have added 2 more fields on it, this will be replicate into to BW system.

This data gets stored in BW system and PSA table holds the data which is extracted from ERP system. PSA tables are stored as a temporary transparent table in BW system and that are created, when we create a mapping of a data-source b/w BW & ERP. PSA is configurable.

Name of the PSA table starts with `/BIO/(Standard)` or `/BIO/(Custom)`.

Pre-requisite:

1. Download the SAP BW ADDON transport request from the Bristlecone site, refer to section "C.2.1.1 Downloading the Transport Request files".

Note: Please use the following link instead mentioned in the referring section:

http://www.bristleconelabs.com/edel/showdownload.html?product=odi_sap_bw_addon.

2. Import the downloaded Transport Request into the source SAP BW system, refer the section "C.2.1.2 Installing the Transport Request Files".

Starting the PSA Reverse-Engineering Process

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

1. In the Models tree view, open the SAP BW SourceModel.
2. In the Reverse Engineer tab:
 - a. Select the Global context.
 - b. Select the Customized option.
 - c. Select the RKM SAP ERP you have imported in the SAP BW Demo project.
 - d. Enter the following parameters in the SAP ERP KM options:
USE_GUI: No
SAP_TABLES_NAME: /BIC/B0000133000
 - e. Set other parameters according to your SAP configuration.
3. From the **File** menu, click **Save** to save your changes.
4. Click **Reverse Engineer** to start the reverse-engineering process.
5. Click **OK**. The Sessions Started Dialog is displayed.
6. Click **OK**.
7. Validate in Operator that the session is now in status running.
If session has failed, validate settings. Do not move on until all installation steps have been completed successfully.
8. Check in Model for the reversed PSA table.



Note:

Currently PSA table reversal is only supported in non-GUI mode.

Designing a Mapping

To create a mapping loading SAP BW data into an Oracle staging area:

1. Create a mapping with source datastores from the SAP BW Model. This mapping should have an Oracle target or use an Oracle schema as the Staging Area.

2. Create joins, filters, and map attributes for your mapping.
3. In the Physical diagram of the mapping, select the access point for the SAP BW source data object(s). The Property Inspector opens for this object.
4. In the Loading Knowledge Module tab, select the LKM SAP BW to Oracle (SQLLDR).

Considerations for SAP BW Integration

This section includes the following topics:

- [File Transfer Configurations](#)
- [Execution Modes](#)
- [Controlling ABAP Uploading / ABAP code in production](#)
- [Managing ODI SAP Transport Requests](#)
- [SAP Packages and SAP Function Groups](#)
- [Log Files](#)
- [Limitation of the SAP BW Adapter](#)

File Transfer Configurations

The ODI SAP adapter extracts data using ABAP programs. For transferring the data from SAP system to the ODI agent the adapter supports two transfer modes and different configurations:

- [Transfer using a Shared Directory \(recommended\)](#)
- [FTP based Transfer](#)

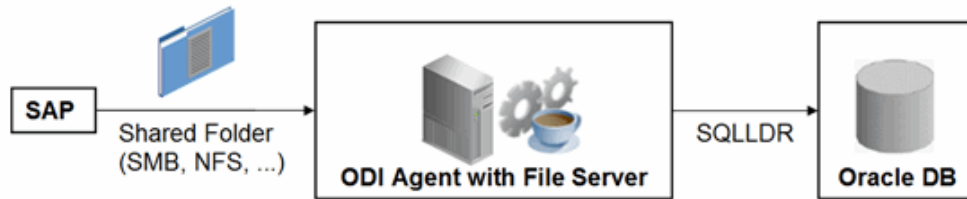
Transfer using a Shared Directory (recommended)

During the extraction process the ABAP programs write chunks of data into the data file in the shared folder. For better performance this shared folder should be located on the ODI agent machine. In this setup, for LKM SAP BW to Oracle (SQLLDR): SQL*Loader reads locally the data file and loads the data into the Oracle staging area. For LKM SAP BW to SQL: The ODI File Driver reads locally the data file and inserts the data using JDBC into a non-Oracle staging area.

If the folder is not located on the ODI agent machine, then the ODI agent first needs to copy the file from the shared folder to the agent for loading the data using SQL*Loader/ JDBC-Connection in the next step.

Configuration 1: Shared Folder is physically located on the ODI Agent machine (recommended)

Figure 7-1 Configuration 1



This configuration is used, when `FTP_TRANSFER_METHOD = FSMOUNT_DIRECT`. In this configuration the following data movements are performed:

1. The ABAP program extracts chunks of `FETCH_BATCH SIZE` records and writes them into a file in the shared folder.
2. for LKM SAP BW to Oracle (SQL*Loader): SQL*Loader reads the data file from this `TEMP_DIR` and loads the data into the Oracle staging area. For LKM SAP BW to SQL: The ODI File Driver reads the data file from this `TEMP_DIR` and inserts the data using JDBC into a non-Oracle staging area.

This configuration requires the following Topology settings:

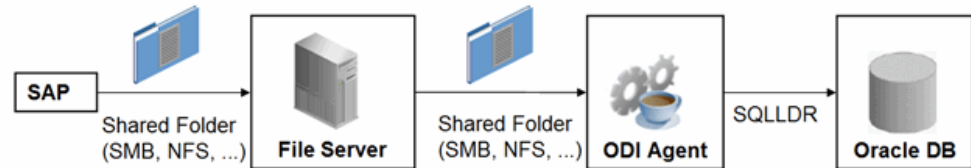
1. Create a File data server pointing to the File server/ODI Agent machine:
 - Host (Data Server): n/a
 - User: n/a
 - Password: n/a
2. Under this File data server create a physical schema representing the shared folder in the File host server. Specify the Data and Work Schemas as follows:
 - *Data Schema*: Path (UNC) of the shared folder used by the ABAP program to write extraction files.
 - *Directory (Work Schema)*: Local path on the server's machine hosting the shared folder. This path is used by an agent and SQL*Loader installed on this machine to access the extraction files without passing via the shared folder.

 **Note:**

Temporary files such as `ctl`, `bad`, `dsc` will be created in a local temporary folder on the run-time agent. The default temporary directory is the system's temporary directory. On UNIX this is typically `/tmp` and on Windows `c:\Documents and Settings\\Local Settings\Temp`. This directory can be changed using the KM option `TEMP_DIR`.

Configuration 2: Shared folder is not physically located on the ODI Agent machine

Figure 7-2 Configuration 2



This configuration is used, when `FTP_TRANSFER_METHOD = FSMOUNT`. In this configuration the following data movements are performed:

1. The ABAP program extracts chunks of `FETCH_BATCH SIZE` records and writes them into a file in the shared folder.
2. The run-time agent copies the file into the directory given by `TEMP_DIR` option of the LKM.
3. for LKM SAP BW to Oracle (SQLLDR): SQL*Loader reads the data file from this `TEMP_DIR` and loads the data into the Oracle staging area. For LKM SAP BW to SQL: The ODI File Driver reads the data file from this `TEMP_DIR` and inserts the data using JDBC into a non-Oracle staging area.

This configuration requires the following Topology settings:

1. Create a File data server pointing to the file server into which the extraction file will be pushed from SAP and picked up from for SQL*Loader.

Set the parameters for this data server as follows:

- Host (Data Server): n/a
 - User: n/a
 - Password: n/a
2. In this File data server create a physical schema representing the shared folder. Specify the Data and Work Schemas as follows:
 - *Directory (Schema)*: Path (UNC) of the shared folder used by the ABAP program to write extraction files, and by the agent to copy the file.
 - *Directory (Work Schema)*: <undefined>. Leave this path blank, as data files are never accessed directly from the File server's file system.

Please note that data files will be copied to the run-time agent from the shared folder in a local temporary folder. The default temporary directory is the system's temporary directory. On UNIX this is typically `/tmp` and on Windows `c:\Documents and Settings \<user>\Local Settings\Temp`. This directory can be changed using the KM option `TEMP_DIR`.

FTP based Transfer

At the end of the extraction process these ABAP programs will upload the data file to a FTP server. For better performance this FTP server should be located on the same machine as the run-time agent.

If the agent is not located on the same machine as the FTP server, it will download the file from the FTP server before loading it to the staging area SQL*Loader/ JDBC-Connection. This download operation is performed using FTP, SFTP or SCP.

Figure 7-3 Configuration 1: FTP Server is installed on an ODI Agent machine



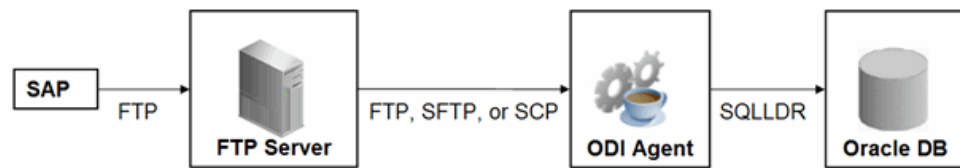
The configuration shown in [Figure 7-3](#) is used, when `FTP_TRANSFER_METHOD = NONE`. In this configuration the following data movements are performed:

1. The ABAP program extracts the data and uploads the data file to the FTP server.
2. For LKM SAP BW to Oracle (SQL*Loader): SQL*Loader reads locally the data file and loads the data into the Oracle staging area. For LKM SAP BW to SQL: The ODI File Driver reads locally the data file and inserts the data using JDBC into a non-Oracle staging area.

This configuration requires the following Topology settings:

1. Create a File data server pointing to the FTP server:
 - Host (Data Server): FTP server host name or IP address.
 - User: Username to log into FTP server.
 - Password: Password for the user.
2. 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:
 - *Directory (Schema)*: Path on the FTP server for uploading SAP extraction files.
 - *Directory (Work Schema)*: Local path on the FTP server's machine containing the SAP extraction file. The agent and SQL*Loader/ODI Flat File Driver read the extraction files from this location.

Figure 7-4 Configuration 2: FTP Server is not installed on ODI Agent machine



The configuration shown in [Figure 7-4](#) is used, when `FTP_TRANSFER_METHOD` is `FTP`, `SFTP` or `SCP`. In this configuration the following data movements are performed:

1. The ABAP program extracts the data and uploads the data file to the FTP server.
2. The ODI agent downloads the file from the FTP server into the directory given by KM Option `TEMP_DIR`.
3. For LKM SAP BW to Oracle (SQLLDR): SQL*Loader reads the data file from this `TEMP_DIR` and loads the data into the Oracle staging area. For LKM SAP BW to SQL: The ODI File Driver reads the data file from this `TEMP_DIR` and inserts the data using JDBC into a non-Oracle staging area.

This configuration requires the following Topology settings:

1. Create a File data server pointing to the FTP server:
 - Host (Data Server): FTP server host name or IP address.
 - User: User name to log into FTP server.
 - Password: Password for the user.
2. 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:
 - *Directory (Schema)*: Path on the FTP server for uploading SAP extraction files.
 - *Directory (Work Schema)*: <undefined>; this path is left blank, as data files are never accessed directly from the FTP server's file system.

Considerations and Limitations:

The FTP based data transfer uses the widely spread (S)FTP file transfer and requires all data to be held in SAP's application server memory before transfer. Therefore the required memory per SAP session increases with the amount of data extracted and will set an upper limit to the data volume. This upper limit can be adjusted to a certain extend by increasing the sessions memory settings in SAP.

The required setup for the shared folder based configuration is slightly more complex, but it removes the need for all data to fit into SAP AS' memory and is therefore the recommended extraction method.

Execution Modes

Background Processing

By default, the generated ABAP code will be deployed as an ABAP report. On execution, this report is submitted to the SAP scheduler for background processing. The job status is then monitored by ODI.

The KM option `JOB_CLASS` defines the priority of the background job. Valid values (corresponding to SAP `JOB_CLASS` settings) are:

`JOB_CLASS = A` is Highest Priority

`JOB_CLASS = B` is Normal Priority

`JOB_CLASS = C` is Lowest Priority

Dialog Mode Processing

For backward compatibility, the KM option `BACKGROUND_PROCESSING` can be set to false. The generated ABAP code will then be deployed as an RFC. On execution, this RFC is called by ODI to extract the data.

Dialog mode processing has been deprecated and is currently supported for backward compatibility only. In future releases, dialog processing may be removed entirely.

Controlling ABAP Uploading / ABAP code in production

During development ODI generates ABAP code and uploads it into the SAP development system with every mapping execution. This automatic code upload allows quick development cycles.

Once a Mapping or Package has been unit tested and is ready to be migrated out of the development environment, the generated SAP ABAP code has to be transported to the respective SAP system using SAP's CTS (Change and Transport System) like any other SAP ABAP code. This is standard SAP practice. To facilitate this task, SAP transport requests are automatically created during upload into development. Please contact your SAP administrator for transporting generated ODI SAP ABAP programs.

In case you are working with distinct ODI repositories for dev, test and production, ensure that your ODI scenario matches the ODI ABAP code of the respective SAP system. That is, you have to transport the SAP ABAP code using SAP CTS from your SAP development system to your SAP QA system *and* transport the ODI scenario (which has generated the transported ABAP code) from your ODI development repository to your ODI QA repository. Please see *Working with Scenarios of Developing Integration Projects with Oracle Data Integrator* for details on how to transport ODI scenarios.

Once outside of development ODI should no longer upload ABAP code, as the ABAP code has been transported by SAP's CTS and such non-development systems usually do not allow ABAP uploading.

Even though uploading can be explicitly turned off by setting the LKM option `UPLOAD_ABAP_CODE` to No, it usually is turned off using the FlexField "SAP Allow ABAP Upload" defined on the SAP data server in ODI Topology: The ABAP code is only uploaded, if both the LKM option `UPLOAD_ABAP_CODE` and the Flexfield `SAP Allow ABAP`

Upload are set to *Yes*. For disabling any upload into production systems it is sufficient to set the Flexfield "SAP Allow ABAP Upload" to 0 in Topology.

 **Tip:**

To configure a mapping that uploads the ABAP code in development but skips the upload in QA or production:

1. Set the KM option `UPLOAD_ABAP_CODE` set to *Yes* in all mappings
2. Configure the SAP data servers in the Topology as follows:
 - Set the Flexfield *SAP Allow ABAP Upload* to 1 for all SAP development systems
 - Set the Flexfield *SAP Allow ABAP Upload* to 0 for all other SAP systems

 **Note:**

Before starting the extraction process, ODI verifies that the mapping/scenario matches the code installed in SAP. If there is a discrepancy - for example, if the scenario was modified but the ABAP code was not re-uploaded - an exception is thrown.

In some situations it may be desirable just to install the Mapping's ABAP extraction code and not to extract any data, such as for an automated installation. In this case all mappings can be linked inside a package with the KM option `EXECUTE_ABAP_CODE` set to *False* in every mapping. Executing this package will then install all ABAP code, but will not perform any execution.

To avoid the modification of all mappings (setting `EXECUTE_ABAP_CODE` to *False* as described above), you can instead disable all SAP ABAP executions by using the FlexField **SAP Allow ABAP Execute** on the ODI DataServer. If this FlexField is *disabled*, the ABAP code is not executed even if the KM option `EXECUTE_ABAP_CODE` is set to *True*.

Manual Upload

In some cases, automatic upload may not be allowed even in development systems. For such situations the KM option `MANUAL_ABAP_CODE_UPLOAD` allows manual uploads. If set to *true*, ODI will create a text file containing the generated ABAP code. Please note that for any ABAP code generation, the DataServer FlexField **SAP Allow ABAP Upload** must be set to 1. This includes manual upload. By default, the text file is created as "REPORT_" + `ABAP_PROG_NAME` + ".ABAP" in the system's temp folder ("java.io.tmpdir"). This code is handed over to the SAP administrator, who will install it. Once installed the ODI mapping can be executed with `MANUAL_ABAP_CODE_UPLOAD` and `UPLOAD_ABAP_CODE` both set back to *false*.

Managing ODI SAP Transport Requests

During development, ABAP code is uploaded to the SAP system with every mapping execution. More precisely:

An ODI mapping extracting SAP data generates one or several ABAP extraction programs (e.g. when join location is set to staging and consequently two extraction jobs are created). By default, all ABAP extraction programs of one mapping are assigned to one SAP function group. The ABAP extraction programs for a different mapping will be assigned to a different SAP function group. The default function group name is similar to `ZODI_FGR_<Mapping Id>`.

During upload, a SAP CTS transport request is created for each ODI Mapping (for each SAP function group). This allows granular deployment of the generated ODI ABAP extraction programs via SAP CTS.

Grouping (Background Processing)

When the ABAP code of multiple ODI Mappings should be grouped into a single transport request for more coarse-grained deployment control, the following steps are needed:

- Set KM option `SAP_REPORT_NAME_PREFIX` for a common prefix for all ABAP reports. For example, `ZODI_DWH_SALES_` on all mappings.
- Choose a KM option `ABAP_PROGRAM_NAME` for every individual mapping. For example, `LOAD01`, `LOAD02`, etc.

These sample settings would result in a single transport request containing the ABAP reports called `ZODI_DWH_SALES_LOAD01`, `ZODI_DWH_SALES_LOAD02`, etc.

Grouping (Dialog mode processing, deprecated)

When the ABAP code of multiple ODI Mappings should be grouped into a single transport request for more coarse-grained deployment control, the KM option `SAP_FUNCTION_GROUP` for all LKMs in these mappings can be set to a user defined value, e.g. `ZODI_FGR_DWH_SALES`. This then leads to ODI generating all ABAP extraction programs into the same SAP function group which is then attached to a single transport request. Please contact your SAP administrator for valid function group names at your site.

Tip:

The name of the generated ABAP extraction programs is by default similar to `ZODI_<Mapping Id>_<SourceSet Id>`. This ensures convenient development due to unique program names. While the MappingId never changes, certain changes to an ODI mapping can cause the SourceSetId to change and consequently cause the respective extraction program name to change. Therefore it is recommended to use user-defined program names, once development stabilizes. ABAP program names can be set by defining a value for LKM option `ABAP_PROGRAM_NAME`, e.g. `ZODI_DWH_SALES_DATA01`. Please contact your SAP administrator for applicable naming conventions.

Transport Request Description

When ODI creates a transport request, it will set the transport request description to the text provided in KM option `SAP_TRANSPORT_REQUEST_DESC` and applies to function group defined in KM option `FUNCTION_GROUP`.

By default, the ODI Step name (which is usually the mapping name) will be used.

Code generation expressions like `ODI:<%=odiRef.getPackage("PACKAGE_NAME")%>` may be useful when grouping several mappings into one SAP function group/ SAP transport request.

SAP Packages and SAP Function Groups

SAP Packages

All SAP objects installed by the ODI SAP Adapter are grouped into SAP packages :

- `ZODI_RKM_PCKG` contains any RKM related objects. These objects are used during development phase.
- `ZODI_LKM_PCKG` contains any extraction programs generated by any SAP LKM.

If requested by the SAP administrator, these default names can be overwritten using the KM options `SAP_PACKAGE_NAME_ODI_DEV` and `SAP_PACKAGE_NAME_ODI_PROD`. These values must be set during the first-time installation. Later changes require a reinstallation of the ODI SAP Adapter with a prior uninstallation.

SAP Function Groups

All SAP function modules are grouped into SAP function groups:

- `ZODI_FGR` contains any function modules needed during development, e.g. for retrieving metadata. These function modules are installed by the RKM during first-time installation.

The default name can be overwritten during first-time RKM installation using RKM option `SAP_FUNCTION_GROUP_ODI_DEV`.

- `ZODI_FGR_PROD` contains any function modules needed at runtime in production, e.g. for monitoring ABAP report execution status. These function modules are installed by the RKM during first-time installation.

The default name can be overwritten during first-time RKM installation using RKM option `SAP_FUNCTION_GROUP_ODI_PROD`.

- `ZODI_FGR_PROD_...` contains any data extraction function modules. Such function modules are generated by the LKM when using deprecated dialog mode (`BACKGROUND_PROCESSING = false`).

By default every mapping uses its own function group. The default values can be overwritten using the LKM option `SAP_FUNCTION_GROUP`, which is independent of the two function group names mentioned above.


See [Managing ODI SAP Transport Requests](#) for more information.

Log Files

During RKM and LKM execution many messages are logged in ODI's logging facility. These messages in the ODI log files and other log files may contain valuable details for troubleshooting. See Runtime Logging for ODI components of *Administering Oracle Data Integrator* for details about the ODI's logging facility. [Table 7-2](#) describes the different log files and their usage:

Table 7-2 Log Files

Default Log File Name	KM / Phase	Content
<ODI Logs>	RKM	Execution Log of metadata retrieval
<ODI Logs>	RKM	Information about first time installation of SAP RFC for RKM
<ODI Logs>	LKM - Generation Time	Information about code generation for ABAP extractor
<ODI Logs>	LKM - Runtime	Information about installation of ABAP extractor
<ODI Logs>	LKM - Runtime	Information about Delta Extraction
<System Temp Dir or local FTP dir>/ ZODI_<Mapping Id>_<SrcSet>_<Context>.log	LKM - Runtime	SQL*Loader log file
<System Temp Dir or local FTP dir>/ ZODI_<Mapping Id>_<SrcSet>_<Context>.out	LKM - Runtime	OS std output during SQL*Loader execution, may contain information, e.g. when SQL*Loader is not installed
<System Temp Dir or local FTP dir>/ ZODI_<Mapping Id>_<SrcSet>_<Context>.err	LKM - Runtime	OS error output during SQL*Loader execution, may contain information, e.g. when SQL*Loader is not installed

 **Not e:**
 Only for RKM based installation

Limitation of the SAP BW Adapter

The SAP ABAP BW adapter has the following limitations:

- The **Test** button for validating SAP Connection definition in ODI's Topology manager is not supported.
- The SAP BW data store type (InfoCube, InfoObject, ODS/DSO, OpenHub, Hierarchy, and Text Table) cannot be changed after a table has been reverse-engineered.
- The SAP ABAP KMs only support Ordered Joins.
- Full Outer join and Right outer joins are not supported.
- In one-to-many relationships (InfoCube and associated InfoObject join), the first data target should be InfoCube and then InfoObjects and its TextTables.
- SAP BW MultiCubes are not supported.

MultiCubes are a logical view on physical InfoCubes and do not hold any data per se. Please consider extracting from base InfoCubes and then join the data on staging area using ODI.

- InfoCubes can be left-outer joined to multiple IOs on source.

 **Note:**

- Only a single one of these InfoObjects can be joined to its Text table on source.
- Joining multiple InfoObjects to their respective Text tables is possible on staging.

- Hierarchy datastores cannot be joined on source with any other SAP BW objects.
- Text datastores of InfoObjects having no master data cannot be joined on source with any other SAP BW objects.
- OpenHub datastores cannot be joined on source with any other SAP BW objects.
- Only attribute RSHIENM can be filtered on using a constant string value, for example `HIER_OGL_ACCOUNT.RSHIENM = 'MYHIER1'`

8

SAP ABAP ERP

Oracle Data Integrator provides specific Knowledge Modules (KMs) for handling SAP ERP data. You can obtain information on how to work with these KMs in this section. This chapter includes the following sections:

- [Introduction](#)
- [Installation and Configuration](#)
- [Defining the Topology](#)
- [Setting up the Project](#)
- [Creating and Reverse-Engineering a Model](#)
- [Designing a Mapping](#)
- [Considerations for SAP ERP Integration](#)

Introduction

The SAP ERP Knowledge Modules let Oracle Data Integrator connect to SAP-ERP system using SAP Java Connector (SAP JCo) libraries. These adapters allows mass data extraction from SAP-ERP systems.

If this is the first time you are using the SAP ERP adapter, it is recommended to review *Getting Started with SAP ABAP ERP Adapter for Oracle Data Integrator*.

It contains the complete pre-requisites list and step-by-step instructions including SAP connection testing.

Concepts

The SAP ERP Knowledge Modules for Oracle Data Integrator use mature integration methods for SAP-ERP system, in order to:

- Reverse-Engineer SAP ERP metadata
- Extract and load data from SAP ERP system (source) to an Oracle or non-Oracle Staging Area

Knowledge Modules

Oracle Data Integrator provides the Knowledge Modules listed in [Table 8-1](#) for handling SAP ERP data.

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

- Reads SAP data from SAP ERP system
- Loads this SAP data into Oracle or non-Oracle Staging Area

- Reverse-engineers SAP Metadata and proposes a tree browser to select only the required Metadata
- Uses flexfields to map the SAP table types (Transparent, Cluster, and Pool)

Table 8-1 SAP ERP KMs

Knowledge Module	Description
LKM SAP ERP to Oracle (SQLLDR)	Extracts data from SAP ERP into a flat file and then loads it into Oracle Staging Area using the SQL*LOADER command line utility.
LKM SAP ERP to SQL	Extracts data from SAP ERP into a flat file and then loads it into a Staging Area using a JDBC connection.
RKM SAP ERP	Reverse-engineering Knowledge Module to retrieve SAP specific metadata for modules, application components, tables, columns, primary keys, foreign keys and indexes.
RKM SAP ERP Connection Test	This RKM is used for testing the SAP connection from Oracle Data Integrator. See SAP Connection Test for more information.

Overview of the SAP ABAP Integration Process

The RKM SAP ERP enables Oracle Data Integrator to connect to SAP ERP system using SAP JCo libraries and perform a customized reverse-engineering of SAP metadata.

The LKM SAP ERP to Oracle (SQLLDR) and LKM SAP ERP to SQL are in charge of extracting and loading data from SAP ERP system (Source) to an Oracle or non-Oracle Staging Area.

Note:

Access to SAP ERP is made through ABAP. As a consequence, the technology used for connecting is SAP ABAP.

Reverse-Engineering Process

Reverse-engineering uses the RKM SAP ERP.

This knowledge module extracts the list of SAP ERP data objects and optionally displays this list in a Metadata Browser graphical interface. The user selects from this list the tables to reverse-engineer.

The reverse-engineering process retrieves tables, primary keys, foreign keys and indexes.

Integration Process

Data integration from SAP is handled by the LKM SAP ERP to Oracle (SQLLDR) and the LKM SAP ERP to SQL.

The LKM SAP ERP to Oracle (SQLLDR) is used for mappings sourcing from SAP via ABAP and having a Staging Area located in an Oracle Database and the LKM SAP ERP to SQL is used for non-Oracle staging areas.

The KM first generates optimized ABAP code corresponding to the extraction process required for a given mapping. This code includes filters and joins that can be processed directly in the source SAP server. This ABAP program is automatically uploaded and is executed using the OdiSAPAbapExecute tool to generate an extraction file in SAP.

The KM then transfers this extraction file either to a pre-configured FTP server or to a shared directory. This file is then either downloaded from this server using FTP, SFTP, SCP or copied to the machine where the ODI Agent is located, and is finally loaded either using SQL*Loader or using a JDBC connection into the staging area. The agent can also directly read the extraction file on the FTP server's disk. See [File Transfer Configurations](#) for more information.

The rest of the integration process (data integrity check and integration) is managed with other Oracle Data Integration KMs.

Installation and Configuration

Make sure you have read the information in this section before you start working with the SAP ERP data:

- [System Requirements and Certification](#)
- [Technology Specific Requirements](#)
- [Connectivity Requirements](#)

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 list of supported platforms and versions is available on Oracle Technical Network (OTN):

<http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html>

Technology Specific Requirements

Some of the Knowledge Modules for SAP ERP use specific features of SAP and of the Oracle database. This section lists the requirements related to these features.

- A JCo version compatible with adapter must be used. The list of supported JCo versions is available from the Oracle Technical Network (OTN). See [System Requirements and Certification](#) for more information.
- A JVM version compatible with both Oracle Data Integrator and JCo must be used.
- The adapter supports two transfer modes for transferring data from SAP system to the ODI agent: data transfer using a Shared Directory and data transfer through FTP. For details and restrictions, see [File Transfer Configurations](#).

Depending on the chosen file transfer mode the following requirements must be met:

– **Data transfer through a Shared Directory (recommended transfer method)**

The LKM SAP ERP to Oracle (SQLLDR) requires a folder that is shared between the SAP system and the ODI agent. The SAP application server transfers the data by writing it out into a folder that is accessible from the SAP system and the ODI agent machine. This is typically done by sharing a folder of the ODI agent machine with the SAP system. Note that the shared folder does not necessarily have to be located on the ODI agent machine. A shared folder on a third machine is also possible, as long as the shared folder is accessible to both the ODI agent machine and the SAP system.

 **Note:**

For security reasons, folders located on the SAP server should not be shared. You should instead share a folder located on the ODI agent machine with the SAP system, or use a third machine as the shared file server.

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

– **Data transfer through FTP**

LKM SAP ERP to Oracle (SQLLDR) requires a FTP server to upload data from the SAP ERP system. This data is either read locally by the agent executing the mapping (when this agent runs on the FTP server machine), or remotely (when this agent is located on a different machine than the FTP server). This FTP server must be accessible over the network from both the SAP ERP machine and the agent machine.

- For "LKM SAP ERP to Oracle (SQLLDR)" only: SQL*Loader is required on the machine running the agent when executing mappings using LKM SAP ERP to Oracle (SQLLDR). SQL*Loader is used for loading data extracted from SAP to the Oracle staging area.

Connectivity Requirements

Oracle Data Integrator connects to the SAP-ERP system hosting the SAP ERP data using JCo. It also uses a FTP Server or a Shared Directory to host the data extracted from the SAP system.

This section describes the required connection information:

- [Installing and Configuring JCo](#)
- [Installing ODI SAP Components into SAP System](#)
- [Requesting FTP Server Access](#)
- [Validating the SAP Environment Setup](#)
- [Gathering SAP Connection Information](#)

- [Gathering FTP Connection Information](#)
- [Gathering Shared Directory Information](#)
- [Adding the Open Tool](#)

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 Technical Network:

<http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html>

Note:

- A minimum version of JCo 3.0.2 is required
 - 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 a temporary directory `<sapjco-install-path>`.
 3. Follow the installation instructions provided in the JCo documentation (`<sapjco-install-path>/javadoc/installation.html`) for your platform.
 4. Copy the required files (`sapjco3.jar` and `sapjco3.dll/.so`) into the `<ODI_HOME>/odi/sdk/lib` directory (ODI Studio, ODI Standalone Agent) and into the `<WLS_DOMAIN>/lib` directory (JEE Agent).
 5. Restart the ODI Components using SAP (ODI Studio, Standalone Agent)
 6. Check the JCo installation.

 **Note:**

For SAP Secure Network Connections, ensure the following:

- the environment variable `SECUDIR` points to the directory containing any certificate files.
- the environment variable `SNC_LIB` points to the directory containing `sapcrypto.dll/libsapcrypto.so` for Linux.

Please contact your SAP Administrator for more details on certificates and Crypto Libraries.

Installing ODI SAP Components into SAP System

The ODI SAP adapter communicates with the SAP System using a few ODI SAP RFCs. These RFCs are installed by your SAP Basis team using SAP Transport requests. Please contact your SAP administrators for installing the ODI SAP Components and assigning the required SAP user authorizations following the instructions given in [Installing ODI SAP Components](#).



Requesting FTP Server Access

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

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

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

To request FTP server access:

1. Login to SAP server.
2. Execute transaction SE16.
3. Enter `SAPFTP_SERVERS` in the **Table Name** field and click  icon to create entries.
4. Enter IP address of the FTP server in the **FTP_SERVER_NAME** field.
5. Enter port of the FTP server in the **FTP_Server_Port** field.
6. Click  icon to save.

Validating the SAP Environment Setup

[Validating the ODI SAP Setup](#) contains instructions for some basic validation of the SAP environment for the use with the ODI SAP Adapter. Please ask your SAP Basis team to run all validations and provide back the validation results like screen shots and confirmations.

Gathering SAP Connection Information

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

- either for SAP Group Logons:
 - SAP System ID: The three-character, unique identifier of a SAP system in a landscape.
 - Message Server: IP address/ host name of SAP Message Server.
 - Message Server Port: port number or service name of SAP Message Server.
 - Group/Server: Name of SAP Logon Group.
- or for SAP Server Logons:
 - SAP ERP System IP Address or Hostname: IP address/ host name of the host on which SAP is running.
 - 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 ERP System IP Address or Hostname: IP address/ Hostname of the host on which SAP is running.
- SAP User: SAP User is the unique user name given to a user for logging on the SAP System.
- SAP Password: Case-sensitive password used by the user to log in.
- SAP Language: Code of the language used when logging in For example: EN for English, DE for German.
- SAP Client Number: The three-digit number assigned to the self-contained unit which is called Client in SAP. A Client can be a training, development, testing or production client or represent different divisions in a large company.
- SAP System Number: The two-digit number assigned to a SAP instance which is also called Web Application Server or WAS.
- SAP System ID: The three-character, unique identifier of a SAP system in a landscape.
- SAP SNC Connection Properties (optional) SAP Router String (optional): SAP is enhancing security through SNC and SAP router. It is used when these securities are implemented.
- SAP Transport Layer Name: This string uniquely identifies a transport layer in a SAP landscape. It allows ODI to create transport requests for later deployment in SAP.
- SAP Temporary Directory (`SAP_TMP_DIR`): This parameter is used to define the custom work directory on the SAP system. It applies only in case of FTP transfer mode. This path is used for temporary files during extraction. The default value

(FlexField value is left empty) is to use the `DIR_HOME` path defined in the SAP profile.

Specify a directory on the SAP application server to be used for temporary files. The path must end on slash/ backslash depending on the OS the SAP application server runs on. The ODI SAP user must have read and write privileges on this directory.

 **Caution:**

The default value (empty FF value) must be avoided for any critical SAP system, as an excessive temp file during extraction could fill up the respective file system and can cause serious issues on the SAP system. Even for SAP development systems it is strongly recommended to override the default value.

- SAP ABAP Version: The version of the SAP system.
- SAP Character Set: The character set is only required if your SAP system is not a UNICODE system. For a complete list of character sets, see [Locale Data](#) in the *Oracle Database Globalization Support Guide*. For example, `EE8ISO8859P2` for Croatian Data. For UNICODE systems, use `UTF8`.

 **Note:**

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

Gathering FTP Connection Information

Gathering FTP connection information only applies if you plan to transfer data using FTP. The SAP ERP system will push data to a server using the FTP protocol. Collect the following information from your system administrator:

- FTP server name or IP address
- FTP login ID
- FTP login password
- Directory path for storing temporary data files

Validate that the FTP server is accessible both from SAP System and from ODI agent machine.

Gathering Shared Directory Information

Gathering Shared Directory information only applies, if you plan to transfer data through a shared directory. The SAP system will push data to a shared folder. For later setup, gather the following information from your system administrator:

- (UNC) path name of the shared folder

Validate that the shared folder is accessible from both the SAP System and the ODI agent machine and does not require any interactive authentication to be accessed.

Please note that the shared folder must be accessible from the SAP system using the <sid>adm user and from the operating system user that starts the ODI agent.

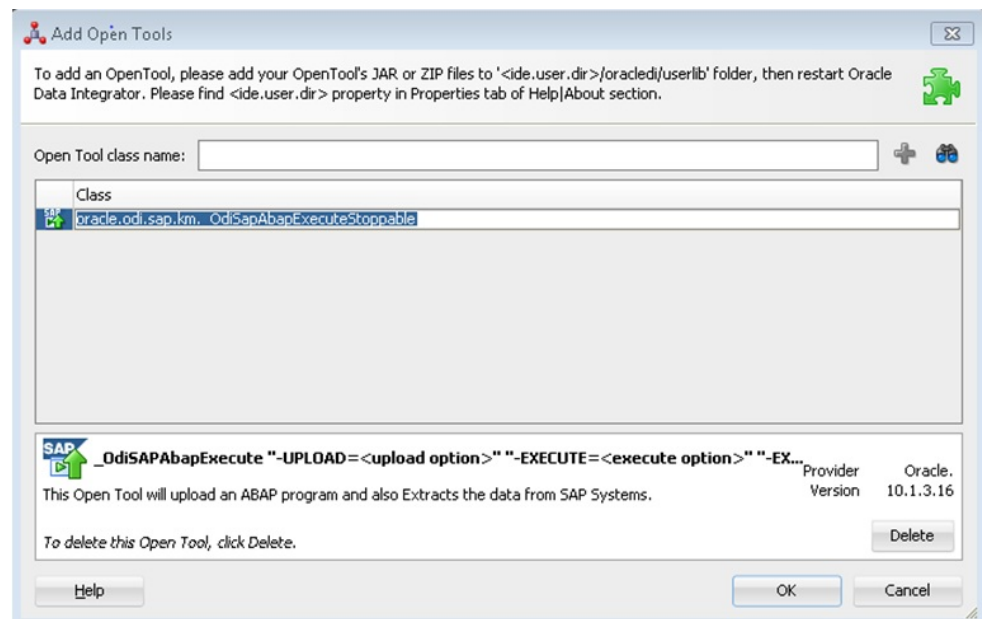
Adding the Open Tool

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

1. Connect to Designer.
2. Select **ODI > Add/Remove Open Tools...**
3. In the Add/remove Open Tools window, enter the following name in the Open Tool class name field:

```
oracle.odi.sap.km._OdiSapAbapExecuteStoppable
```

4. Click **Add Open Tool**.
5. The Open Tool appears as shown below.



6. Click **OK**.

Defining the Topology

You must define the two data servers used for SAP integration. The SAP ABAP Data Server and the File Data server.

1. [Create the File Data Server](#)
2. [Create the SAP ABAP Data Server](#)

Create the File Data Server

This data server corresponds to the FTP server or File Server into which the extraction file will be pushed from SAP and picked up for Loading.

Create a File Data Server

Create a File data server as described in *Creating a File Data Server of the Connectivity and Knowledge Modules Guide for Oracle Data Integrator*. This section describes the parameters specific to SAP ERP.

Depending on the chosen data transfer mode, this data server must point either to:

- An existing FTP server into which the extraction file will be pushed from SAP and picked up for Loading or
- The shared folder into which the SAP system will write the extraction file and from which SQL*Loader/ ODI Flat File Driver will pick it up. This schema represents the folder in the FTP host where the extraction file will be pushed.

Note that the parameters for the data server depend on the data transfer mode.

- When transferring data through FTP, set the parameters 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
- When transferring data through a Shared Directory, set the parameters as follows:
 - Host (Data Server): n/a
 - User: n/a
 - Password: n/a
- For use with "LKM SAP ERP to SQL" these additional parameters must be configured:
 - JDBC driver class: com.sunopsis.jdbc.driver.file.FileDriver
 - JDBC URL: jdbc:snps:dbfile?ENCODING=UTF8

The above URL is for SAP UNICODE systems. For non-UNICODE systems, refer details on ENCODING parameter in *Creating a File Data Server of the 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.

See [File Transfer Configurations](#) for more information.

Create the File Schema

In this File data server create a Physical Schema as described in *Creating a File Physical Schema of the Connectivity and Knowledge Modules Guide for Oracle Data Integrator*.

This schema represents the folder that is located either in the FTP host or the File Server. Note that this is the folder into which the extraction file will be pushed. Depending on the data transfer mode, specify the Data and Work Schemas as follows:

- For the FTP file transfer:
 - *Directory (Schema)*: Path on the FTP server to upload or download extraction files from the remote location. This path is used when uploading extraction files from the SAP ERP system into the FTP server. It is also used by a remote agent to download the extraction files. Note that this path must use slashes and must end with a slash character.
 - *Directory (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 FTP. This access method is used if the `FTP_TRANSFER_METHOD` parameter of the LKM SAP ERP to Oracle (SQLLDR)/ LKM SAP ERP to SQL is set to `NONE`. The Work Schema is a local directory location. As a consequence, slashes or backslashes should be used according to the operating system. This path must end with a slash or backslash.

Path names given on Data and Work schemas are not necessarily the same: the FTP server may provide access to a FTP directory named `/sapfiles` - the value for *Directory (Schema)* - while the files are accessed locally in `c:\inetpub\ftproot\sapfiles` - the value for *Directory (Work Schema)*.

- For the Shared Directory transfer:
 - *Directory (Schema)*: Path (UNC) of the shared folder to write and read extraction files. SAP System writes the extraction files into this folder. It is also used by a remote agent to copy the extraction files to the ODI agent machine. Note that this path must use slashes or backslashes according to the operating system of the SAP Application Server and must end with a slash or backslash character.
 - *Directory (Work Schema)*: Local path on the server's machine hosting the shared folder. This path is used by an agent installed on this machine to access the extraction files without passing through the shared folder. This access method is used if the `FTP_TRANSFER_METHOD` parameter of the LKM SAP ERP to Oracle (SQLLDR)/ LKM SAP ERP to SQL is set to `FSMOUNT_DIRECT`. The Work Schema is a local directory location. As a consequence, slashes or backslashes should be used according to the operating system. This path must end with a slash or backslash.

See [File Transfer Configurations](#) for more information.

Create a File Logical Schema called `File Server for SAP ABAP`, and map it to the Physical Schema. The name of this Logical Schema name is predefined and must be `File Server for SAP ABAP`.

Create the SAP ABAP Data Server

This SAP ABAP data server corresponds to the SAP server from which data will be extracted.

Create the SAP ABAP Data Server

To configure a SAP ABAP data server:

1. Create a data server for the SAP ABAP technology using the standard procedure, as described in Creating a Data Server of the *Developing Integration Projects with Oracle Data Integrator*. This data server uses the SAP connection information.
2. Set connection parameters:
 - Name: The name of the data server as it will appear in ODI. For example, `SAP_ERP`
 - User: SAP ERP User, as provided by the SAP Administrator
 - Password: This user's SAP ERP Password. This password is case-sensitive.
3. Set Logon type specific connection parameters:
 - either for SAP Group Logons (on FlexField tab)
 - FlexField SAP System ID: The three-character, unique identifier of a SAP system in a landscape.
 - FlexField SAP Message Server Host: IP address/ host name of SAP Message Server.
 - FlexField SAP Message Server Port: port number or service name of SAP Message Server.
 - FlexField SAP Group: Name of SAP Logon Group.
 - or for SAP Server Logons
 - Host (Data Server): SAP ERP System IP Address or host name.
 - FlexField 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.
 - FlexField SAP System Number: The two-digit number assigned to a SAP instance which is also called Web Application Server or WAS.
 - FlexField SAP System ID: The three-character, unique identifier of a SAP system in a landscape.
4. Set SAP Secure Network Settings (optional):
 - FlexField "for SNC: Level of Security": 0 for "No SNC", 1 for "Authentication", 2 for "Integrity", 3 for "Encryption", 9 for "Maximum Security Settings Available"
 - FlexField "for SNC: Client Name": SNC Client Name
 - FlexField "for SNC: Partner Name": SNC Partner Name
5. Set other flexfields 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 Router String: Router String. This parameter is optional and can be left empty.
 - SAP Character Set: Character set of the SAP system. This information is used for SQLLDR when loading data. For UNICODE SAP systems, use the default value `UTF8`. For a complete list of character sets, see [Locale Data](#) in the *Oracle Database Globalization Support Guide*. For example, `EE8ISO8859P2` for Croatian Data.

- SAP Temporary Directory: In case of FTP based file transfer (compared to shared folder transfer) data is extracted to the default temp directory of the SAP server. If needed, a different folder can be used instead.
- SAP Temporary Directory Name (Logical) (optional): While **SAP Temporary Directory** specifies a *physical* SAP directory, the FlexField **SAP Temporary Directory (logical)** can be used to specify a *logical* SAP directory. During the ABAP execution the logical path is translated into the physical path by the SAP system.

 **Note:**

Please note the following:

- Only either **SAP Temporary Directory Name** or **SAP Temporary Directory Name (Logical)** can be specified.
- The physical path defined in the SAP system must end on [back]slash.

- SAP Extract Timeout: Maximum number of seconds ODI will wait for a SAP background job before it times out.
- SAP ABAP Version: Enter the SAP ABAP version as follows:
 - For SAP 4.6C enter 46C
 - For SAP 4.7 enter 620
 - For SAP ECC 5.0 enter 640
 - For SAP ECC 6.0 enter 700
- SAP ERP Version: Reserved for future use.
- SAP Allow ABAP Upload: set to 1, if ABAP code can be uploaded on this SAP System. Typically set to 0 for any non-development system. See [Controlling ABAP Uploading / ABAP code in production](#) for more details.
- SAP Allow ABAP Execute: set to 1, if ABAP code can be executed on this SAP System. See [Controlling ABAP Uploading / ABAP code in production](#) for more details.

 **Note:**

The Test button for validating the SAP Connection and the FTP Connection definition is not supported.

- SAP Transport Layer Name: This transport layer is used by ODI to create any new transport requests. The default transport layer name is SAP. If the SAP system uses a different transport layer, this FlexField must be updated accordingly. Otherwise any TR creation will fail.

Except for Data Server Name, all the parameters that you provide while defining the SAP Data Server should be provided by the SAP Administrators. See [Gathering SAP Connection Information](#) for more information about these parameters.

Create the SAP ABAP Schema

To configure a SAP ABAP schema:

1. Create a Physical Schema under the SAP ABAP data server as described in *Creating a Physical Schema* in *Administering Oracle Data Integrator*. This schema does not require any specific configuration. Only one physical schema is required under a SAP ABAP data server.
2. Create a Logical Schema for this Physical Schema as described in *Creating a Logical Schema* in *Administering Oracle Data Integrator* in the appropriate context.

Setting up the Project

Setting up a project using SAP ERP features follows the standard procedure. See *Creating an Integration Project* of the *Developing Integration Projects with Oracle Data Integrator*.

Import the following KMs into your Oracle Data Integrator project:

- RKM SAP ERP
- RKM SAP ERP Connection Test
- LKM SAP ERP to Oracle (SQLLDR)
- LKM SAP ERP to SQL

In addition to these specific SAP ERP KMs, import the standard Oracle LKMs, IKMs, and CKMs to perform data extraction and data quality checks with an Oracle database. See *Oracle Database* in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for a list of available KMs.

Creating and Reverse-Engineering a Model

This section contains the following topics:

- [Creating a SAP ERP Model](#)
- [Reverse-Engineering a SAP ERP Model](#)

Creating a SAP ERP Model

Create an SAP ERP Model based on the SAP ABAP technology and on the SAP ABAP logical schema using the standard procedure, as described in *Creating a Model* of the *Developing Integration Projects with Oracle Data Integrator*.

Reverse-Engineering a SAP ERP Model

To perform a Customized Reverse-Engineering with the RKM SAP ERP, use the usual procedure, as described in *Reverse-engineering a Model* of *Developing Integration Projects with Oracle Data Integrator*. This section details only the fields specific to SAP ERP:

1. In the Reverse Engineer tab of the SAP ERP Model, select the RKM SAP ERP.
2. For the RKM SAP ERP, set the USE_GUI KM option to `true`.

3. Save the model.
4. Click **Reverse-Engineer** in the Model Editor toolbar.
5. The Tree Metadata Browser appears after the session is started. Select the table(s) to reverse.
6. Click **Reverse-Engineer** in the Tree Metadata Browser window.

The reverse-engineering process returns the selected tables as datastores.

 **Note:**

If the reverse-engineering is executed on a run-time agent, the `USE_GUI` option should be set to false. This option should be used only when the customized reverse-engineering is started using the agent built-in the Studio.

Designing a Mapping

To create a mapping loading SAP ERP data into an Oracle staging area:

1. Create a mapping with source datastores from the SAP ERP Model. This mapping should have an Oracle target or use an Oracle schema as the Staging Area.
2. Create joins, filters, and map attributes for your mapping.
3. In the Physical diagram of the mapping, select the access point for the SAP ABAP source table(s). The Property Inspector opens for this object.
4. In the Loading Knowledge Module tab, select the LKM SAP ERP to Oracle (SQLLDR)/LKM SAP ERP to SQL.

Considerations for SAP ERP Integration

This section includes the following topics:

- [File Transfer Configurations](#)
- [Execution Modes](#)
- [Controlling ABAP Uploading / ABAP code in production](#)
- [Managing ODI SAP Transport Requests](#)
- [SAP Packages and SAP Function Groups](#)
- [Log Files](#)
- [Limitations of the SAP ABAP Adapter](#)

File Transfer Configurations

The ODI SAP adapter extracts data using ABAP programs. For transferring the data from SAP system to the ODI agent the adapter supports two transfer modes and different configurations:

- [Transfer using a Shared Directory \(recommended\)](#)

- FTP based Transfer

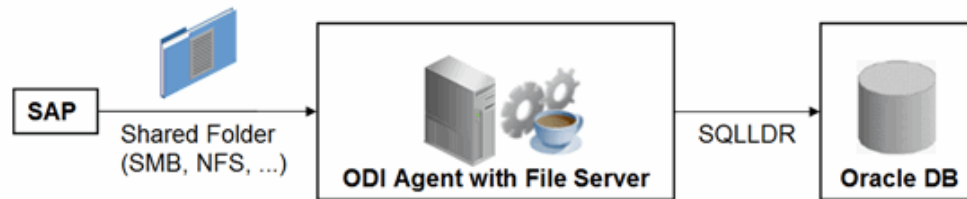
Transfer using a Shared Directory (recommended)

During the extraction process the ABAP programs write chunks of data into the data file in the shared folder. For better performances this shared folder should be located on the ODI agent machine. In this setup, SQL*Loader directly reads the data from this folder and loads it into database.

If the folder is not located on the ODI agent machine, then the ODI agent first needs to copy the file from the shared folder to the agent for loading the data using SQL*Loader/ JDBC-Connection in the next step.

Configuration 1: Shared Folder is physically located on the ODI Agent machine (recommended)

Figure 8-1 Configuration 1



This configuration is used, when `FTP_TRANSFER_METHOD = FSMOUNT_DIRECT`. In this configuration the following data movements are performed:

1. The ABAP program extracts chunks of `FETCH_BATCH SIZE` records and writes them into a file in the shared folder.
2. for LKM SAP ERP to Oracle (SQL*Loader): SQL*Loader reads locally the data file and loads the data into the Oracle staging area.

For LKM SAP ERP to SQL: The ODI File Driver reads locally the data file and inserts the data using JDBC into a non-Oracle staging area.

This configuration requires the following Topology settings:

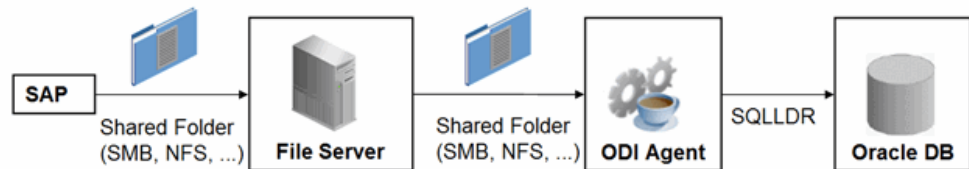
1. Create a File data server pointing to the File server/ODI Agent machine:
 - Host (Data Server): n/a
 - User: n/a
 - Password: n/a
2. Under this File data server create a physical schema representing the shared folder in the File host server. Specify the Data and Work Schemas as follows:
 - *Data Schema*: Path (UNC) of the shared folder used by the ABAP program to write extraction files.
 - *Directory (Work Schema)*: Local path on the server's machine hosting the shared folder. This path is used by an agent and SQL*Loader installed on this machine to access the extraction files without passing via the shared folder.

 **Note:**

Temporary files such as ctl, bad, dsc will be created in a local temporary folder on the run-time agent. The default temporary directory is the system's temporary directory. On UNIX this is typically /tmp and on Windows c:\Documents and Settings\\Local Settings\Temp. This directory can be changed using the KM option TEMP_DIR.

Configuration 2: Shared folder is not physically located on the ODI Agent machine

Figure 8-2 Configuration 2



This configuration is used, when `FTP_TRANSFER_METHOD = FSMOUNT`. In this configuration the following data movements are performed:

1. The ABAP program extracts chunks of `FETCH_BATCH SIZE` records and writes them into a file in the shared folder.
2. The run-time agent copies the file into the directory given by `TEMP_DIR` option of the LKM.
3. for LKM SAP ERP to Oracle (SQL*Loader): SQL*Loader reads the data file from this `TEMP_DIR` and loads the data into the Oracle staging area.

For LKM SAP ERP to SQL: The ODI File Driver reads the data file from this `TEMP_DIR` and inserts the data using JDBC into a non-Oracle staging area.

This configuration requires the following Topology settings:

1. Create a File data server pointing to the file server into which the extraction file will be pushed from SAP and picked up from for SQL*Loader.

Set the parameters for this data server as follows:

- Host (Data Server): n/a
 - User: n/a
 - Password: n/a
2. In this File data server create a physical schema representing the shared folder. Specify the Data and Work Schemas as follows:
 - *Directory (Schema)*: Path (UNC) of the shared folder used by the ABAP program to write extraction files, and by the agent to copy the file.
 - *Directory (Work Schema)*: <undefined>. Leave this path blank, as data files are never accessed directly from the File server's file system.

Please note that data files will be copied to the run-time agent from the shared folder in a local temporary folder. The default temporary directory is the system's temporary directory. On UNIX this is typically `/tmp` and on Windows `c:\Documents and Settings\\Local Settings\Temp`. This directory can be changed using the KM option `TEMP_DIR`.

FTP based Transfer

During the extraction the ABAP program writes out data into a temporary file. This file is created in directory defined by the ODI DataServer FlexField `SAP_TMP_DIR`. See section [Gathering SAP Connection Information](#) for more details about `SAP_TMP_DIR`.

At the end of the extraction process these ABAP programs will upload the data file to a FTP server. For better performances this FTP server should be located on the same machine as the run-time agent.

If the agent is not located on the same machine as the FTP server, it will download the file from the FTP server before loading it to the staging area using `SQL*Loader / JDBC-Connection`. This download operation is performed using `FTP, SFTP or SCP`.

Figure 8-3 Configuration 1: FTP Server is installed on an ODI Agent machine



The configuration shown in [Figure 8-3](#) is used, when `FTP_TRANSFER_METHOD = NONE`. In this configuration the following data movements are performed:

1. The ABAP program extracts the data and uploads the data file to the FTP server.
2. For LKM SAP ERP to Oracle (SQLLDR): `SQL*Loader` reads locally the data file and loads the data into the Oracle staging area.

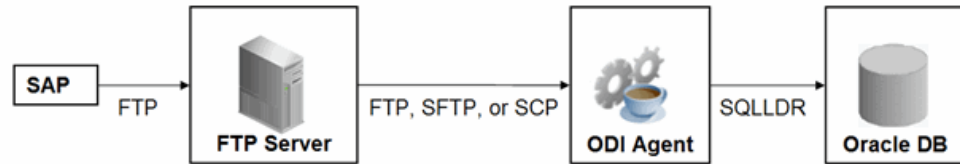
For LKM SAP ERP to SQL: The ODI File Driver reads locally the data file and inserts the data using `JDBC` into a non-Oracle staging area.

This configuration requires the following Topology settings:

1. Create a File data server pointing to the FTP server:
 - Host (Data Server): FTP server host name or IP address.
 - User: Username to log into FTP server.
 - Password: Password for the user.
2. 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:

- *Directory (Schema)*: Path on the FTP server for uploading SAP extraction files.
- *Directory (Work Schema)*: Local path on the FTP server's machine containing the SAP extraction file. The agent and SQL*Loader/ODI Flat File Driver read the extraction files from this location.

Figure 8-4 Configuration 2: FTP Server is not installed on ODI Agent machine



The configuration shown in [Figure 8-4](#) is used, when `FTP_TRANSFER_METHOD` is `FTP`, `SFTP` or `SCP`. In this configuration the following data movements are performed:

1. The ABAP program extracts the data and uploads the data file to the FTP server.
2. The ODI agent downloads the file from the FTP server into the directory given by KM Option `TEMP_DIR`.
3. For LKM SAP ERP to Oracle (SQLLDR): SQL*Loader reads the data file from this `TEMP_DIR` and loads the data into the Oracle staging area.

For LKM SAP ERP to SQL: The ODI File Driver reads the data file from this `TEMP_DIR` and inserts the data using JDBC into a non-Oracle staging area.

This configuration requires the following Topology settings:

1. Create a File data server pointing to the FTP server:
 - Host (Data Server): FTP server host name or IP address.
 - User: User name to log into FTP server.
 - Password: Password for the user.
2. 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:
 - *Directory (Schema)*: Path on the FTP server for uploading SAP extraction files.
 - *Directory (Work Schema)*: <undefined>; this path is left blank, as data files are never accessed directly from the FTP server's file system.

Considerations and Limitations:

The FTP based data transfer uses the widely spread (S)FTP file transfer and requires all data to be held in SAP's application server memory before transfer. Therefore the required memory per SAP session increases with the amount of data extracted and will set an upper limit to the data volume. This upper limit can be adjusted to a certain extend by increasing the sessions memory settings in SAP.

The required setup for the shared folder based configuration is slightly more complex, but it removes the need for all data to fit into SAP AS' memory and is therefore the recommended extraction method.

Execution Modes

Background Processing

By default, the generated ABAP code will be deployed as an ABAP report. On execution, this report is submitted to the SAP scheduler for background processing. The job status is then monitored by ODI.

The KM option `JOB_CLASS` defines the priority of the background job. Valid values (corresponding to SAP `JOB_CLASS` settings) are as follows:

`JOB_CLASS = A` is Highest Priority

`JOB_CLASS = B` is Normal Priority

`JOB_CLASS = C` is Lowest Priority

Dialog mode processing

For backward compatibility, the KM option `BACKGROUND_PROCESSING` can be set to `false`. The generated ABAP code will then be deployed as an RFC. On execution, this RFC is called by ODI to extract the data.

Dialog mode processing has been deprecated and is currently supported for backward compatibility only. In future releases, dialog processing may be removed entirely.

Controlling ABAP Uploading / ABAP code in production

During development ODI generates ABAP code and uploads it into the SAP development system with every mapping execution. This automatic code upload allows quick development cycles.

Once a Mapping or Package has been unit tested and is ready to be migrated out of the development environment, the generated SAP ABAP code has to be transported to the respective SAP system using SAP's CTS (Change and Control System) like any other SAP ABAP code. This is standard SAP practice. To facilitate this task, SAP transport requests are automatically created during upload into development. Please contact your SAP administrator for transporting generated ODI SAP ABAP programs.

In case you are working with distinct ODI repositories for dev, test and production, ensure that your ODI scenario matches the ODI ABAP code of the respective SAP system. That is, you have to transport the SAP ABAP code using SAP CTS from your SAP development system to your SAP QA system *and* transport the ODI scenario (which has generated the transported ABAP code) from your ODI development repository to your ODI QA repository. Please see *Working with Scenarios of Developing Integration Projects with Oracle Data Integrator* for details on how to transport ODI scenarios.

Once outside of development ODI should no longer upload ABAP code, as the ABAP code has been transported by SAP's CTS and such non-development systems usually do not allow ABAP uploading.

Even though uploading can be explicitly turned off by setting the LKM option `UPLOAD_ABAP_CODE` to `No`, it usually is turned off using the FlexField "SAP Allow ABAP Upload" defined on the SAP data server in ODI Topology: The ABAP code is only uploaded, if both the LKM option `UPLOAD_ABAP_CODE` and the Flexfield `SAP Allow ABAP`

Upload are set to *Yes*. For disabling any upload into production systems it is sufficient to set the Flexfield "SAP Allow ABAP Upload" to 0 in Topology.

 **Tip:**

To configure a mapping that uploads the ABAP code in development but skips the upload in QA or production:

1. Set the KM option `UPLOAD_ABAP_CODE` set to *Yes* in all mappings
2. Configure the SAP data servers in the Topology as follows:
 - Set the Flexfield *SAP Allow ABAP Upload* to 1 for all SAP development systems
 - Set the Flexfield *SAP Allow ABAP Upload* to 0 for all other SAP systems

 **Note:**

Before starting the extraction process, ODI verifies that the mapping/scenario matches the code installed in SAP. If there is a discrepancy - for example, if the scenario was modified but the ABAP code was not re-uploaded - an exception is thrown.

In some situations it may be desirable to install the Mapping's ABAP extraction code and not to extract any data, such as for an automated installation. In this case all mappings can be linked inside a package with the KM option `EXECUTE_ABAP_CODE` set to *False* in every mapping. Executing this package will then install all ABAP code, but will not perform any execution.

To avoid the modification of all mappings (setting `EXECUTE_ABAP_CODE` to *False* as described above), you can instead disable all SAP ABAP executions by using the FlexField **SAP Allow ABAP Execute** on the ODI DataServer. If this FlexField is *disabled*, the ABAP code is not executed even if the KM option `EXECUTE_ABAP_CODE` is set to *True*.

Manual Upload

In some cases automatic upload may not be allowed even in development systems. For such situations the KM option `MANUAL_ABAP_CODE_UPLOAD` allows manual uploads. If set to *true*, ODI will create a text file containing the generated ABAP code. Please note that for any ABAP code generation, the DataServer FlexField **SAP Allow ABAP Upload** must be set to 1. This includes manual upload. By default, the text file is created as "REPORT_" + `ABAP_PROG_NAME` + ".ABAP" in the system's temp folder ("java.io.tmpdir"). This code is handed over to the SAP administrator, who will install it. Once installed the ODI mapping can be executed with `MANUAL_ABAP_CODE_UPLOAD` and `UPLOAD_ABAP_CODE` both set back to *false*.

Managing ODI SAP Transport Requests

During development, ABAP code is uploaded to the SAP system with every mapping execution. More precisely:

An ODI mapping extracting SAP data generates one or several ABAP extraction programs (e.g. when join location is set to staging and consequently two extraction jobs are created). By default all ABAP extraction programs of one mapping are assigned to one SAP function group. The ABAP extraction programs for a different mapping will be assigned to a different SAP function group. The default function group name is similar to `ZODI_FGR_<Mapping Id>`.

During upload a SAP CTS transport request is created for each ODI Mapping (for each SAP function group). This allows granular deployment of the generated ODI ABAP extraction programs via SAP CTS.

Grouping (Background processing)

When the ABAP code of multiple ODI Mappings should be grouped into a single transport request for more coarse-grained deployment control, the following steps are needed:

- Set KM option `SAP_REPORT_NAME_PREFIX` for a common prefix for all ABAP reports. For example, `ZODI_DWH_SALES_` on all mappings.
- Choose a KM option `ABAP_PROGRAM_NAME` for every individual mapping. For example, `LOAD01`, `LOAD02`, etc.

These sample settings would result in a single transport request containing the ABAP reports called `ZODI_DWH_SALES_LOAD01`, `ZODI_DWH_SALES_LOAD02`, etc.

Grouping (Dialog mode processing, deprecated)

When the ABAP code of multiple ODI Mappings should be grouped into a single transport request for more coarse-grained deployment control, the KM option `SAP_FUNCTION_GROUP` for all LKMs in these mappings can be set to a user defined value, e.g. `ZODI_FGR_DWH_SALES`. This then leads to ODI generating all ABAP extraction programs into the same SAP function group which is then attached to a single transport request. Please contact your SAP administrator for valid function group names at your site.

Tip:

The name of the generated ABAP extraction programs is by default similar to `ZODI_<Mapping Id>_<SourceSet Id>`. This ensures convenient development due to unique program names. While the MappingId never changes, certain changes to an ODI mapping can cause the SourceSetId to change and consequently cause the respective extraction program name to change. Therefore it is recommended to use user-defined program names, once development stabilizes. ABAP program names can be set by defining a value for LKM option `ABAP_PROGRAM_NAME`, e.g. `ZODI_DWH_SALES_DATA01`. Please contact your SAP administrator for applicable naming conventions.

Transport Request Description

When ODI creates a transport request, it will set the transport request description to the text provided in KM option `SAP_TRANSPORT_REQUEST_DESC` and applies to function group defined in KM option `FUNCTION_GROUP`.

By default the ODI Step name (which is usually the mapping name) will be used.

Code generation expressions like `ODI:<%=odiRef.getPackage("PACKAGE_NAME")%>` may be useful when grouping several mappings into one SAP function group/ SAP transport request.

SAP Packages and SAP Function Groups

SAP Packages

All SAP objects installed by the ODI SAP Adapter are grouped into SAP packages:

- `ZODI_RKM_PCKG` contains any RKM related objects. These objects are used during development phase.
- `ZODI_LKM_PCKG` contains any extraction programs generated by any SAP LKM.

If requested by the SAP administrator, these default names can be overwritten using the KM options `SAP_PACKAGE_NAME_ODI_DEV` and `SAP_PACKAGE_NAME_ODI_PROD`. These values must be set during the first-time installation. Later changes require a reinstallation of the ODI SAP Adapter with a prior uninstallation.

Please note that LKM option `SAP_PACKAGE_NAME` must always use the same value given during first-time RKM installation. This means that when non-default values are used during first-time RKM installation, **all** mappings must set the LKM option `SAP_PACKAGE_NAME` to the non-default value.

SAP Function Groups

All SAP function modules are grouped into SAP function groups:

- `ZODI_FGR` contains any function modules needed during development, e.g. for retrieving metadata. These function modules are installed by the RKM during first-time installation.

The default name can be overwritten during first-time RKM installation using RKM option `SAP_FUNCTION_GROUP_ODI_DEV`.

- `ZODI_FGR_PROD` contains any function modules needed at runtime in production, e.g. for monitoring ABAP report execution status. These function modules are installed by the RKM during first-time installation.

The default name can be overwritten during first-time RKM installation using RKM option `SAP_FUNCTION_GROUP_ODI_PROD`.

- `ZODI_FGR_PROD_...` contains any data extraction function modules. Such function modules are generated by the LKM when using deprecated dialog mode (`BACKGROUND_PROCESSING=false`).

By default every mapping uses its own function group. The default values can be overwritten using the LKM option `SAP_FUNCTION_GROUP`, which is independent of the two function group names mentioned above.


See [Managing ODI SAP Transport Requests](#) for more information.

Log Files

During RKM and LKM execution many messages are logged in ODI's logging facility. These messages in the ODI log files and other log files may contain valuable details for troubleshooting. See Runtime Logging for ODI components of *Administering Oracle Data Integrator* for details about the ODI's logging facility. [Table 8-2](#) describes the different log files and their usage:

Table 8-2 Log Files

Default Log File Name	KM / Phase	Content
<ODI Logs>	RKM	Execution Log of metadata retrieval
<ODI Logs>	RKM	Information about first time installation of SAP RFC for RKM
<ODI Logs>	LKM - Generation Time	Information about code generation for ABAP extractor
<ODI Logs>	LKM - Runtime	Information about installation of ABAP extractor
<System Temp Dir or local FTP dir>/ ZODI_<Mapping Id>_<SrcSet>_<Context>.log	LKM - Runtime	SQL*Loader log file
<System Temp Dir or local FTP dir>/ ZODI_<Mapping Id>_<SrcSet>_<Context>.out	LKM - Runtime	OS std output during SQL*Loader execution, may contain information, e.g. when SQL*Loader is not installed
<System Temp Dir or local FTP dir>/ ZODI_<Mapping Id>_<SrcSet>_<Context>.err	LKM - Runtime	OS error output during SQL*Loader execution, may contain information, e.g. when SQL*Loader is not installed

 **Not e:**
Only for RKM based installation.

Limitations of the SAP ABAP Adapter

The SAP ABAP adapter has the following limitations:

- The **Test** button for validating SAP Connection definition in ODI's Topology manager is not supported.
- The SAP table type (Transparent, Pool, and Cluster) cannot be changed after a table has been reverse-engineered.
- The SAP ABAP KMs only support Ordered Joins.
- Full Outer join and Right outer joins are not supported.
- In one-to-many relationships, the first table of a join must be the one-table. For example, when joining MARA and MARC, MARA must be the first table in the join.
- Mapping expression executed on the source must not contain any transformations.

A

Additional Information for SAP ABAP BW Adapter

For connecting to the SAP system, you need to configure certain privileges. Also, it is possible to test the connection outside of ODI using a standalone java utility. This appendix includes the following sections:

- [SAP ABAP BW Required Privileges](#)
- [SAP Stand-Alone Connection Test](#)

For more information about the SAP BW KMs, see *Connectivity and Knowledge Modules Guide for Oracle Data Integrator Developer's Guide*.

SAP ABAP BW Required Privileges

For connecting to the SAP system a SAP dialog user is required for 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 \(Background & Foreground\)](#)
- [Authorizations Required for LKM SAP BW Execution \(Background & Foreground\)](#)

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



Note:

Developer Key is required in this authorization.

Object	Field Name	Value (BI 7.0)	Value (BI 7.5)
S_RFC	ACTVT	16	16
S_RFC	RFC_NAME	RFC1, RSAB, RSB3RD, SDIFRUNTIME, SUTL, SYST, SYSU, ZODI*	RFC1, RSAB, RSB3RD, SDIFRUNTIME, SUTL, SYST, SYSU, ZODI*, RSBAPI_IOBJ
S_RFC	RFC_TYPE	FUGR	FUGR
S_RFCRAIAR	RFC_RAIAR	NR	16
S_TCODE	TCD	SU53, SE38	SU53, SE38
S_ADMI_FCD	S_ADMI_FCD	RFCA	RFCA
S_CTS_ADMI	CTS_ADMFCT	TABL	TABL
S_DATASET	ACTVT	06, 33, 34	06, 33, 34, A7
S_DATASET	FILENAME	*	*
S_DATASET	PROGRAM	SAPLSLOG, SAPLSTRF, SAPLZODI*	*
S_TABU_DIS	ACTVT	3	2, 3
S_TABU_DIS	DICBERCLS	*	*
S_DEVELOP	ACTVT	01, 02, 03, 06, 07, 16, 70, MA	01, 02, 03, 06, 07, 16, 70, MA

Object	Field Name	Value (BI 7.0)	Value (BI 7.5)
S_DEVELOP	DEVCLASS	\$TMP, ZODI_LKM_PCKG, ZODI_RKM_PCKG, RSS, ZODI*	\$TMP, ZODI_LKM_PCKG, ZODI_RKM_PCKG, RSS, ZODI*
S_DEVELOP	OBJNAME	*	*
S_DEVELOP	OBJTYPE	FUGR, PROG, DEVC, DEBUG	*
S_DEVELOP	P_GROUP	*	*
S_TRANSPRT	ACTVT	01, 02, 03	01, 02, 03
S_TRANSPRT	TTYPE	DTRA, TASK	DTRA, TASK
S_RS_ADMWB	ACTVT	03, 16	03, 16
S_RS_ADMWB	RSADMWBOBJ	INFOBJECT	INFOBJECT
S_RS_ICUBE	ACTVT	03	03
S_RS_ICUBE	RSICUBEOBJ	*	*
S_RS_ICUBE	RSINFOAREA	*	*
S_RS_ICUBE	RSINFOCUBE	*	*
S_RS_IOBJ	ACTVT	03	03
S_RS_IOBJ	RSIOBJ	*	*
S_RS_IOBJ	RSIOBJCAT	*	*
S_RS_IOBJ	RSIOBJPART	DEFINITION	DEFINITION
S_RS_ODSO	ACTVT	03	03
S_RS_ODSO	RSINFOAREA	*	*
S_RS_ODSO	RSODSOBJ	*	*
S_RS_ODSO	RSODSPART	DATA, DEFINITION	DATA, DEFINITION

Authorizations Required for RKM SAP BW Execution

The following authorizations are required for **RKM SAP BW** execution.

Object	Field Name	Value (BI 7.0)	Value (BI 7.5)
S_RFC	ACTVT	16	16
S_RFC	RFC_NAME	RFC1, RSAB, RSB3RD, SDIFRUNTIME, SUTL, SYST, SYSU, ZODI*	RFC1, RSAB, RSB3RD, SDIFRUNTIME, SUTL, SYST, SYSU, ZODI*, RSBAPI_IOBJ
S_RFC	RFC_TYPE	FUGR	FUGR
S_TCODE	TCD	SU53	SU53
S_DATASET	ACTVT	06, 33, 34	06, 33, 34
S_DATASET	FILENAME	*	*

Object	Field Name	Value (BI 7.0)	Value (BI 7.5)
S_DATASET	PROGRAM	SAPLSTRF, SAPLSLOG, SAPLZODI*	SAPLSTRF, SAPLSLOG, SAPLZODI*
S_TABU_DIS	ACTVT	03	03
S_TABU_DIS	DICBERCLS	*	*
S_TRANSPRT	ACTVT	01, 02, 03	01, 02, 03
S_TRANSPRT	TTYPE	DTRA, TASK	DTRA, TASK
S_RS_ADMWB	ACTVT	03,16	03,16
S_RS_ADMWB	RSADMWBOBJ	*	*
S_RS_ICUBE	ACTVT	03	03
S_RS_ICUBE	RSICUBEOBJ	DATA, DEFINITION	DATA, DEFINITION
S_RS_ICUBE	RSINFOAREA	*	*
S_RS_ICUBE	RSINFOCUBE	*	*
S_RS_IOBJ	ACTVT	03	03
S_RS_IOBJ	RSIOBJ	*	*
S_RS_IOBJ	RSIOBJCAT	*	*
S_RS_IOBJ	RSIOBJPART	DEFINITION	DEFINITION
S_RS_ODSO	ACTVT	03	03
S_RS_ODSO	RSINFOAREA	*	*
S_RS_ODSO	RSODSOBJ	*	*
S_RS_ODSO	RSODSPART	DATA, DEFINITION	DATA, DEFINITION
S_RS_AUTH	BIAUTH	0*	0*

Authorizations Required for LKM SAP BW Upload (Background & Foreground)

The following authorizations are required for **LKM SAP BW** execution in a SAP development environment.



Note:

Developer Key is required in this authorization.

This section contains authorizations for both background and foreground executions.

Object	Field Name	Value (BI 7.0)	Value (BI 7.5)
S_RFC	ACTVT	16	16

Object	Field Name	Value (BI 7.0)	Value (BI 7.5)
S_RFC	RFC_NAME	RFC1, SDIFRUNTIME, SLST, SYST, ZODI*	RFC1, SDIFRUNTIME, SLST, SYST, ZODI*
S_RFC	RFC_TYPE	FUGR	FUGR
S_TCODE	TCD	SE38,SU53	SE38,SU53
S_DEVELOP	ACTVT	1, 2, 3, 6, 7, 16	1, 2, 3, 6, 7, 16
S_DEVELOP	DEVCLASS	*	*
S_DEVELOP	OBJNAME	*	*
S_DEVELOP	OBJTYPE	FUGR, FUNC, PROG*, SUSO	FUGR, FUNC, PROG*, SUSO
S_DEVELOP	P_GROUP	NR	NR
S_PROGRAM	P_ACTION	BTCSUBMIT, SUBMIT, VARIANT	BTCSUBMIT, SUBMIT, VARIANT
S_PROGRAM	P_GROUP	NR	NR
S_TRANSPRT	ACTVT	1, 2, 43	1, 2, 43
S_TRANSPRT	TTYPE	DTRA, TASK	DTRA, TASK
S_TABU_DIS	ACTVT	3	3
S_TABU_DIS	DICBERCLS	*	*
S_BTCH_ADM	BTCADMIN	Y	Y
S_BTCH_JOB	JOBACTION	RELE	RELE
S_BTCH_JOB	JOBGROUP	''	''
S_CTS_ADMI	CTS_ADMFCT	TABL	TABL
S_DATASET	ACTVT	06,33,34	06,33,34
S_DATASET	FILENAME	*	*
S_DATASET	PROGRAM	*	*
S_RS_AUTH	BIAUTH	*	*
S_RS_ICUBE	ACTVT	3, 6	3, 6
S_RS_ICUBE	RSICUBEOBJ	DATA, DEFINITION	DATA, DEFINITION
S_RS_ICUBE	RSINFOAREA	*	*
S_RS_ICUBE	RSINFOCUBE	*	*
S_RS_IOBJ	ACTVT	3	3
S_RS_IOBJ	RSIOBJ	*	*
S_RS_IOBJ	RSIOBJCAT	*	*
S_RS_IOBJ	RSIOBJPART	DATA, DEFINITION	DATA, DEFINITION
S_RS_ODSO	ACTVT	3	3
S_RS_ODSO	RSINFOAREA	*	*
S_RS_ODSO	RSODSOBJ	*	*

Object	Field Name	Value (BI 7.0)	Value (BI 7.5)
S_RS_ODSO	RSODSPART	DATA, DEFINITION	DATA, DEFINITION
S_APPLOG	ACTVT	03, 06	03, 06
S_APPLOG	ALG_OBJECT	*	*
S_APPLOG	ALG_SUBOBJ	*	*

Authorizations Required for LKM SAP BW Execution (Background & Foreground)

The following authorizations are required for **LKM SAP BW** execution in a SAP production environment.

This section contains authorizations for both background and foreground executions.

Object	Field Name	Value (BI 7.0)	Value (BI 7.5)
S_RFC	ACTVT	16	16
S_RFC	RFC_NAME	RFC1, SDIFRUNTIME, SLST, SYST, ZODI_FGR_PROD*	RFC1, SDIFRUNTIME, SLST, SYST, ZODI_FGR_PROD*
S_RFC	RFC_TYPE	FUGR	FUGR
S_TCODE	TCD	SU53	SU53
S_TABU_DIS	ACTVT	3	3
S_TABU_DIS	DICBERCLS	*	*
S_BTCH_ADM	BTCADMIN	Y	Y
S_BTCH_JOB	JOBACTION	RELE	RELE
S_BTCH_JOB	JOBGROUP	''	''
S_DATASET	ACTVT	06,33,34	06, 33, 34
S_DATASET	FILENAME	*	*
S_DATASET	PROGRAM	*	*
S_TRANSPRT	ACTVT	1, 2, 43	1, 2, 43
S_TRANSPRT	TTYPE	DTRA, TASK	DTRA, TASK
S_RS_AUTH	BIAUTH	*	*
S_RS_ICUBE	ACTVT	3, 6	3, 6, 23, 63, 66
S_RS_ICUBE	RSICUBEOBJ	*	*
S_RS_ICUBE	RSINFOAREA	*	*
S_RS_ICUBE	RSINFOCUBE	*	*
S_RS_IOBJ	ACTVT	3	3
S_RS_IOBJ	RSIOBJ	*	*

Object	Field Name	Value (BI 7.0)	Value (BI 7.5)
S_RS_IOBJ	RSIOBJCAT	*	*
S_RS_IOBJ	RSIOBJPART	DATA, DEFINITION	DATA, DEFINITION
S_RS_ODSO	ACTVT	3	3
S_RS_ODSO	RSINFOAREA	*	*
S_RS_ODSO	RSODSOBJ	*	*
S_RS_ODSO	RSODSPART	DATA, DEFINITION	DATA, DEFINITION
S_APPLOG	ACTVT	03, 06	03, 06
S_APPLOG	ALG_OBJECT	*	*
S_APPLOG	ALG_SUBOBJ	*	*

SAP Stand-Alone Connection Test

In addition to the Connection Testing described in the *Getting Started with SAP ABAP BW Adapter for Oracle Data Integrator*, a test can be performed outside of ODI using a standalone java utility. This test is the same for SAP ERP and SAP BW.

See [SAP Connection Test](#) for more information.

SAP Stand-Alone Connection Test

In addition to the Connection Testing, a test can be performed outside of ODI using a standalone java utility. This utility is available with the Standalone agent.

To use a standalone java utility to test the connection:

1. Open a command window.
2. Go to the <ODI_HOME>/odi/sdk/lib directory.
3. Make sure that JAVA_HOME points to a supported JVM.
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 command generates a ODI_SAP_CON_POOL.jcoDestination file in the folder <ODI_HOME>/odi/sdk/lib folder.

For more information, see Appendix C, SAP Stand-Alone Connection Test in the *Getting Started with SAP ABAP BW Adapter for Oracle Data Integrator*.

6. Use a text editor to open the ODI_SAP_CON_POOL.jcoDestination file. This file should look as follows:

```
#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.
8. Launch the utility using the same command. The utility uses the file that you have edited, and outputs the test results or the possible issues.

In addition to just testing the SAP connection, the utility will also validate the existence of certain Function Modules required for the RKM.

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

B

Additional Information for SAP ABAP ERP Adapter

For connecting to the SAP system, you need to configure certain privileges. Also, it is possible to test the connection outside of ODI using a standalone java utility. This appendix includes the following sections:

- [SAP ABAP ERP Required Privileges](#)
- [SAP Connection Test](#)
- [SAP Stand-Alone Connection Test](#)

For more information about the SAP ERP KMs, see *Connectivity and Knowledge Modules Guide for Oracle Data Integrator Developer's Guide*.

SAP ABAP ERP Required Privileges

An SAP dialog user is required for connecting to the SAP system, for the LKM execution during development. A SAP RFC user is *not* sufficient.

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

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

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

Table B-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

Authorizations for ECC6 Servers

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

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

Important points to consider

Consider the following points while configuring the SAP privileges:

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

Authorizations Required for RKM SAP ERP Upload

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



Note:

Developer key is required in this authorization.

Object	Field Name	Value (ECC6-EHP3)	Value (ECC6-EHP7)
S_RFC	ACTVT	16	16
S_RFC	RFC_NAME	RFC1, SDIFRUNTIME, SUTL, SYST, ZODI_FGR_PROD*, ZODI_FGR	RFC1, SDIFRUNTIME, SUTL, SYST, ZODI_FGR_PROD*, ZODI_FGR
S_RFC	RFC_TYPE	FUGR	FUGR
S_TCODE	TCD	SU53, SE38	SE38, WE30, SU53
S_ADMI_FCD	S_ADMI_FCD	NR	RFCA
S_CTS_ADMI	CTS_ADMFCT	TABL	TABL
S_CTS_SADM	CTS_ADMFCT	NR	TABL
S_CTS_SADM	DESTSYS	NR	*
S_CTS_SADM	DOMAIN	NR	*
S_DATASET	ACTVT	34	33, 34
S_DATASET	FILENAME	*	*
S_DATASET	PROGRAM	SAPLSTRF	*
S_TABU_DIS	ACTVT	03	02, 03
S_TABU_DIS	DICBERCLS	*	*
S_TABU_NAM	ACTVT	NR	02, 03
S_TABU_NAM	TABLE	NR	*
S_DEVELOP	ACTVT	01, 02, 03	01, 02, 03, 06

Object	Field Name	Value (ECC6-EHP3)	Value (ECC6-EHP7)
S_DEVELOP	DEVCLASS	\$TMP, ZODI_LKM_PCKG, ZODI_RKM_PCKG	\$TMP, ZODI_LKM_PCKG, ZODI_RKM_PCKG
S_DEVELOP	OBJNAME	Z\$\$\$XRFC, ZODI*	Z\$\$\$XRFC, ZODI*
S_DEVELOP	OBJTYPE	DEVC, FUGR, FUNC, PROG	DEVC, FUGR, FUNC, PROG
S_DEVELOP	P_GROUP		' '
S_PROGRAM	P_ACTION	NR	BTCSSUBMIT, SUBMIT, VARIANT
S_PROGRAM	P_GROUP	NR	*
S_SYS_RWBO	ACTVT	NR	01, 02, 03, 05, 06, 23
S_SYS_RWBO	DESTSYS	NR	*
S_SYS_RWBO	DOMAIN	NR	' '
S_SYS_RWBO	TTYPE	NR	DTRA
S_TRANSPRT	ACTVT	01,03	01, 02, 03, 06
S_TRANSPRT	TTYPE	DTRA, TASK	*
S_IDOCDEFT	ACTVT	NR	03
S_IDOCDEFT	EDI_CIM	NR	
S_IDOCDEFT	EDI_DOC	NR	
S_IDOCDEFT	EDI_TCD	NR	

Authorizations Required for RKM SAP ERP Execution

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

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

Object	Field Name	Value (ECC6–EHP3)	Value (ECC6–EHP7)
S_TABU_NAM	TABLE	NR	*
S_TRANSPRT	ACTVT	01, 03	01, 03
S_TRANSPRT	TTYPE	DTRA, TASK	DTRA, TASK

Authorizations Required for LKM SAP ERP Upload

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



Note:

Developer Key is required in this authorization.

Object	Field Name	Value (ECC6-EHP3)	Value (ECC6-EHP7)
S_RFC	ACTVT	16	16
S_RFC	RFC_NAME	RFC1, SDIFRUNTIME, SYST, ZODI_FGR, ZODI_FGR_PROD_*	RFC1, SDIFRUNTIME, SYST, ZODI_FGR, ZODI_FGR_PROD_*
S_RFC	RFC_TYPE	FUGR	FUGR
S_TCODE	TCD	SE38, SU53	SE38, SU53
S_ADMI_FCD	S_ADMI_FCD	RFCA, SFTP	RFCA, SFTP
S_BTCH_ADM	BTCADMIN	Y	Y
S_BTCH_JOB	JOBACTION	RELE	RELE
S_BTCH_JOB	JOBGROUP	' '	' '
S_CTS_ADMI	CTS_ADMFCT	TABL	TABL
S_DATASET	ACTVT	06, 33, 34	06, 33, 34
S_DATASET	FILENAME	*	*
S_DATASET	PROGRAM	*	SAPLSLOG, SAPLSTRF, SAPLZ*, ZODI*
S_DEVELOP	ACTVT	1	1
S_DEVELOP	DEVCLASS	*	*
S_DEVELOP	OBJNAME	ZODI*	ZODI*
S_DEVELOP	OBJTYPE	FUGR, FUNC	FUGR, FUNC
S_DEVELOP	P_GROUP	NR	
S_TRANSPRT	ACTVT	01, 02	01, 02
S_TRANSPRT	TTYPE	DTRA, TASK	DTRA, TASK
S_APPL_LOG	ACTVT	6	6

Object	Field Name	Value (ECC6-EHP3)	Value (ECC6-EHP7)
S_APPL_LOG	ALG_OBJECT	ZODI_APPOBJ	ZODI_APPOBJ
S_APPL_LOG	ALG_SUBOBJ	ZODI_SUBOBJ	ZODI_SUBOBJ
S_TABU_DIS	ACTVT	3	3
S_TABU_DIS	DICBERCLS	*	*

Authorizations Required for LKM SAP ERP Execution

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

Object	Field Name	Value (ECC6-EHP3)	Value (ECC6-EHP7)
S_RFC	ACTVT	16	16
S_RFC	RFC_NAME	RFC1, SDIFRUNTIME, SYST, ZODI_FGR_PROD*	RFC1, SDIFRUNTIME, SYST, ZODI_FGR_PROD*
S_RFC	RFC_TYPE	FUGR	FUGR
S_ADMI_FCD	S_ADMI_FCD	SFTP	SFTP
S_BTCH_ADM	BTCADMIN	Y	Y
S_BTCH_JOB	JOBACTION	RELE	RELE
S_BTCH_JOB	JOBGROUP	' '	' '
S_DATASET	ACTVT	06, 33, 34	06, 33, 34
S_DATASET	FILENAME	*	*
S_DATASET	PROGRAM	SAPLSLOG*, SAPLSTRF*, SAPLZODI*, ZODI*, SAPLZODI_FGR_PROD*	SAPLSLOG*, SAPLSTRF*, SAPLZODI*, SAPLZODI_FGR_PROD*, ZODI*
S_TRANSPRT	ACTVT	01, 02	01, 02
S_TRANSPRT	TTYPE	DTRA, TASK	DTRA, TASK
S_APPL_LOG	ACTVT	6	6
S_APPL_LOG	ALG_OBJECT	ZODI_APPOBJ	ZODI_APPOBJ
S_APPL_LOG	ALG_SUBOBJ	ZODI_SUBOBJ	ZODI_SUBOBJ

SAP Connection Test

This connection test should be performed after configuring the SAP ABAP data server in the topology.

This test performs the following operations:

- It establishes a test connection to the configured SAP system.
- It validates the proper setup of SAP JCo by displaying the About dialog.

This test is performed using an RKM called RKM SAP ERP Connection Test. This KM does not perform any reverse operation. It simply runs the connection test.

To run the connection test:

1. Create an SAP ERP Model based on the SAP ABAP technology and on the SAP ABAP logical schema using the standard procedure, as described in *Creating a Model of the Developing Integration Projects with Oracle Data Integrator*.
2. In the Reverse Engineer tab of the SAP ERP Model, select the RKM SAP ERP Connect Test.
3. Save the model.
4. Click **Reverse-Engineer** to start the reverse.

Within a few seconds, the SAP JCo About dialog should appear. If not, review the session execution log in the Operator Navigator. Please verify that the output of task "Test SAP Connection" contains `Successfully connected to SAP System`. If not, the connection test has failed and the connection problem must be analyzed.

SAP Stand-Alone Connection Test

In addition to the Connection Testing, a test can be performed outside of ODI using a standalone java utility. This utility is available with the Standalone agent.

To use a standalone java utility to test the connection:

1. Open a command window.
2. Go to the `<ODI_HOME>/odi/sdk/lib` directory.
3. Make sure that `JAVA_HOME` points to a supported JVM.
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 command generates a `ODI_SAP_CON_POOL.jcoDestination` file in the folder `<ODI_HOME>/odi/sdk/lib` folder.

For more information, see Appendix C, SAP Stand-Alone Connection Test in the *Getting Started with SAP ABAP ERP Adapter for Oracle Data Integrator*.

6. Use a text editor to open the `ODI_SAP_CON_POOL.jcoDestination` file. This file should look as follows:

```
#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.
8. Launch the utility using the same command. The utility uses the file that you have edited, and outputs the test results or the possible issues.

In addition to just testing the SAP connection, the utility will also validate the existence of certain Function Modules required for the RKM.

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

C

Components, Compatibility, and Upgrading of the ODI SAP Adapter

It is important to understand the components and compatibility of the ODI SAP Adapter. You can also upgrade to a new ODI SAP Adapter version if required. This appendix includes the following sections:

- [Components](#)
- [Compatibility](#)
- [Upgrading the ODI SAP Adapter](#)
- [Upgrading from ODI 11g to ODI 12c](#)

Components

The ODI SAP ABAP Adapter consists of three components:

- ODI SAP LKMs and ODI SAP RKMs
- ODI SAP OpenTool (`odi-sap.jar`)
- ODI SAP Components

The KMs connect to the SAP system either via the SAP JCo or via the ODI SAP OpenTool (which internally also uses the SAP JCo). So either directly or indirectly the KMs connect to the SAP system and call RFCs delivered as part of the ODI SAP Components.

Note:

Please note the following points:

- KMs are centrally installed in the ODI repository and are shipped with ODI Software or patches.
- The ODI SAP OpenTool (`odi-sap.jar`) is part of each local ODI software installation. This applies to ODI Studio installations and ODI Agent installations. The OpenTool is part of the ODI Software.
- The ODI SAP Components must be installed in every SAP system which ODI is to connect to. This installation is done as part of the Installation of the ODI SAP Adapter.

Compatibility

The following are a few rules regarding compatibility:

- Newer KMs often require a newer OpenTool version. The required minimum OpenTool version is given in the KM description. The OpenTool version can be checked by executing the `odi-sap.jar` using `java -jar odi-sap.jar`.
- Newer KMs often require newer ODI SAP Components. The need for updating is described in the respective release or patch note.
- ODI OpenTools are backward compatible: any OpenTool can be upgraded to a newer version without impacting existing ODI mappings or ODI Scenarios.
- Existing ODI Scenarios work with newer ODI SAP Component versions and newer OpenTool versions.

Upgrading the ODI SAP Adapter

Note:

The primary source for upgrade steps are product or patch release notes. There may be other steps required than just the ones described below.

If you have any doubts, contact Oracle Product Support.

Unless otherwise specified, upgrading to a new ODI SAP Adapter version consists of the following steps:

- **Upgrading the OpenTool**

There should not be any need to manually upgrade the OpenTool, as it is shipped as part of the ODI Software or Patch. If needed, the OpenTool is upgraded, by replacing the `odi-sap.jar`.

- **Upgrading the ODI SAP Components**

Upgrading the ODI SAP Components is described in [Updating ODI SAP Components](#).

- **Upgrading KMs**

In case the SAP Adapter update has happened as part of an ODI Update, KMs may have already been replaced by the Upgrade Assistant, if **Replace KMs with Mandatory Updates** had been checked. If not, there are two methods for manually updating the KMs in a repository. Please choose based on your requirements. Both methods may be combined, for example, start with method 1 for a few mappings and if tests have been successful, method 2 can be applied.

Note:

No matter which KM update method is used, it is recommended to test all mappings.

Method 1: Add as new KMs

This update path leaves the existing KMs unmodified and imports the updated KMs as new KMs. All existing mappings and existing ODI Scenarios will stay unchanged. This approach allows a step-by-step update to the new KMs, but requires every mapping / model to be updated to use the new KMs.

- Import all new KM files in **Duplication** mode inside your project or as global KMs depending on what you have been using so far. This will create new KM objects.
- Rename each KM from Copy of <KM Name> to <KM Name>. For example, Copy of LKM SAP ERP to SQL to LKM SAP ERP to SQL v38.
- For every mapping / model (which should use the updated KMs) switch the L/RKM to v38, for example LKM SAP ERP to SQL to LKM SAP ERP to SQL v38.

 **Note:**

Existing ODI Scenarios will stay unchanged, until they are regenerated.

Replace existing KMs

This update path replaces the active KMs inside the repository with the new versions. Any mapping or model using the old KMs will immediately be using the updated KM. Existing ODI Scenarios will stay unchanged, until they are regenerated.

Steps to repeat for every KM:

- Take a backup of existing KM. For example, right-click the KM and click **Export**.
- Right-click the KM, select **Import Replace...**, select the corresponding new KM file and then click **OK**.
- No changes required to any mappings or models.

 **Note:**

Existing ODI Scenarios will stay unchanged, until they are regenerated.

Upgrading from ODI 11g to ODI 12c

ODI 12c introduces GUIDs for identifying repository objects (as opposed to internal ids as in ODI 11g) and mappings (as opposed to ODI 11g interfaces). The introduction of GUIDs means that any odiRef API call using internal ids will fail, unless the 12c repository is in legacy mode. The 11g ODI SAP scenarios use internal ids and thus will only work in the 12c legacy repository mode. Another implication of moving to GUIDs is that the default ABAP_PROGRAM_NAME will change during upgrade. Please note that this default ABAP_PROGRAM_NAME is intended only for convenience during development. See [Managing ODI SAP Transport Requests](#) for more information.

**Note:**

Any explicitly set `ABAP_PROGRAM_NAME` KM option will remain unchanged.

Evolving from 11g interfaces to 12c mappings means that during upgrade any 11g interface will be converted into a new 12c mapping object. This new mapping will still represent the very same data transformation, but its new 12c semantics will result in a different ABAP code being generated even when using the very same SAP KM version.

Upgrade Path

The upgrade path keeps all the existing ODI SAP scenarios running in ODI 11g for a start (phase 1). In a later step-by-step migration, the ODI SAP scenarios are regenerated/redeployed and then execution for these scenarios is switched from ODI 11g to ODI 12c (phase 2). The upgrade path is in line with general ODI upgrade recommendations.

This upgrade path has the following advantages:

- No ABAP redeploy needed in phase 1
- No ODI scenario regeneration needed in phase 1
- Use of latest SAP KMs (latest fixes/features) for every phase-2-upgraded mapping

The only inconvenience would be the parallel use of ODI 11g and ODI 12c (completely separate repositories and agents).

Phase 1: Use ODI 11g

In the first phase, we preserve the existing ODI 11g repository and execution environment and set up a new parallel ODI 12c environment. This phase consists of the following steps:

- Upgrading from ODI 11g to ODI 12c using all default options (Replace KMs) following the standard ODI upgrade instructions.
- Preserving the ODI 11g environment. Do not run any in-place upgrade, as you will continue using the ODI 11g environment.
- Continuing use of ODI 11g for running any ODI SAP jobs.

Phase 2: Migrate to 12c

Over time, all ODI SAP interfaces/scenarios need to be regenerated and redeployed. Once an interface/mapping/scenario has been redeployed using ODI 12c, the execution of the respective 11g interface/mapping/scenario should be disabled (e.g. deactivate in 11g scheduler), as the job is now run from ODI 12c. Phase 2 consists of the following steps:

- Following upgrade instructions for ODI SAP Components. See [Updating ODI SAP Components](#) for more information. This step only needs to be executed for the very first SAP interface being upgraded.
- Regenerating ODI SAP mapping/scenario.

- Redeploying generated SAP ABAP code using `ABAP_UPLOAD = true`. See [Controlling ABAP Uploading / ABAP code in production](#) for more information.
- Following standard procedures to deploy ODI scenarios into production. For SAP specific details, please see [Moving ODI and SAP Components from Development to Production](#).
- Deactivating execution from ODI 11g environment and activating execution from ODI 12c environment (e.g. by updating schedulers).

Once all ODI SAP mappings/scenarios have been migrated, the ODI 11g environment can be switched off.

D

Installing ODI SAP Components

It is important to understand how to update, install, and uninstall ODI SAP components. You can also validate the ODI SAP setup to ensure that previous installation steps were successful.

This appendix includes the following sections:

- [Updating ODI SAP Components](#)
- [Installing ODI SAP Components](#)
- [Validating the ODI SAP Setup](#)
- [Uninstalling ODI SAP Components](#)

Updating ODI SAP Components

The update path for the ODI SAP Components depends on which ODI SAP Adapter version had been used for installation.

- Installations of ODI SAP Adapter prior to v32

For ODI SAP Adapter versions prior to v32 (= SAP package name is ZODIBW_PCKG, ZODIERP_PCKG, or ZODI_DEVCLASS, or the SAP function group is ZODI_FGR, then please contact Oracle support for details on upgrading, particularly on how to uninstall the existing ODI SAP Components and how to handle any SAP function modules and reports for existing ODI SAP mappings or interfaces.

To check the SAP package name, log in to SAP system >Go to transaction SE80> provide package name as above.

- Installations of ODI SAP Adapter v32 or later

For any SAP systems, which have been installed using the ODI SAP Adapter v32 or later, there is no need for uninstallation. The new ODI SAP Components can be installed on top of an existing installation following the instructions in [Installing ODI SAP Components](#).

If you have any queries, please contact Oracle Technical Support.

Installing ODI SAP Components

Installation of the ODI SAP components is done through the use of TRs (SAP Transport Requests). These transport requests also contain the SAP user profiles. This section describes how to import the ODI SAP components and grant the required authorizations to the ODI SAP user by assigning the ODI SAP profiles.

This section includes the following topics:

- [Installing SAP Transport Request \(TR\)](#)
- [Installing and Assigning SAP User Profile](#)

Installing SAP Transport Request (TR)

Installation of SAP Transport Request requires downloading the TRs from Bristlecone e-delivery website.

This section includes the following topics:

- [Downloading the Transport Request files](#)
- [Installing the Transport Request Files](#)
- [Creating Transport Request for ODI SAP Production Objects](#)
- [Creating Transport Request for ODI SAP Development Objects](#)

Downloading the Transport Request files

Perform the following steps to download the TR files:

1. Go to the following Download URL:

http://www.bristleconelabs.com/edel/showdownload.html?product=odi_sap_km_transport

2. Select the required file and click **Download**. You will be redirected to Login Page.

If you are a registered user, log in with the registered Email Id as user name and the password received.

If you are a new user, perform the following steps to register:

- a. Go to the following URL:

<http://www.bristleconelabs.com/edel/Register.html>

- b. Click **Register**.
- c. Fill in the details in the registration form.

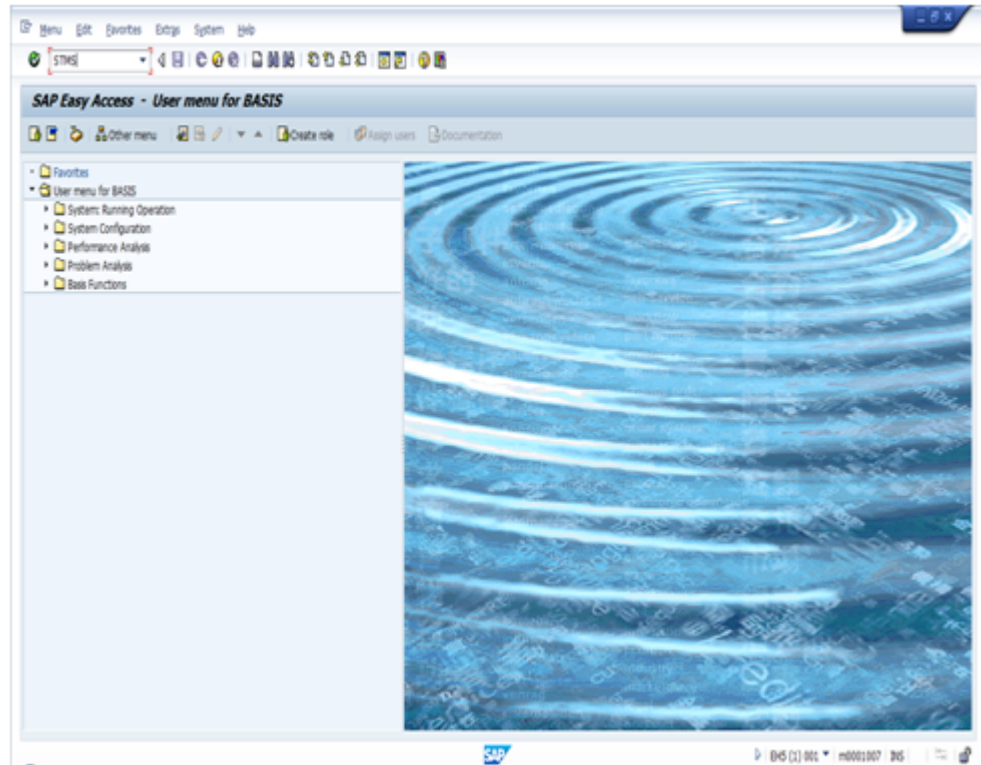
In the **Email ID** field, provide a valid business email address. This will be used to email your password.

- d. Reopen the download link mentioned in step 1.
- e. Select the appropriate SAP version of the TR and download.


Installing the Transport Request Files

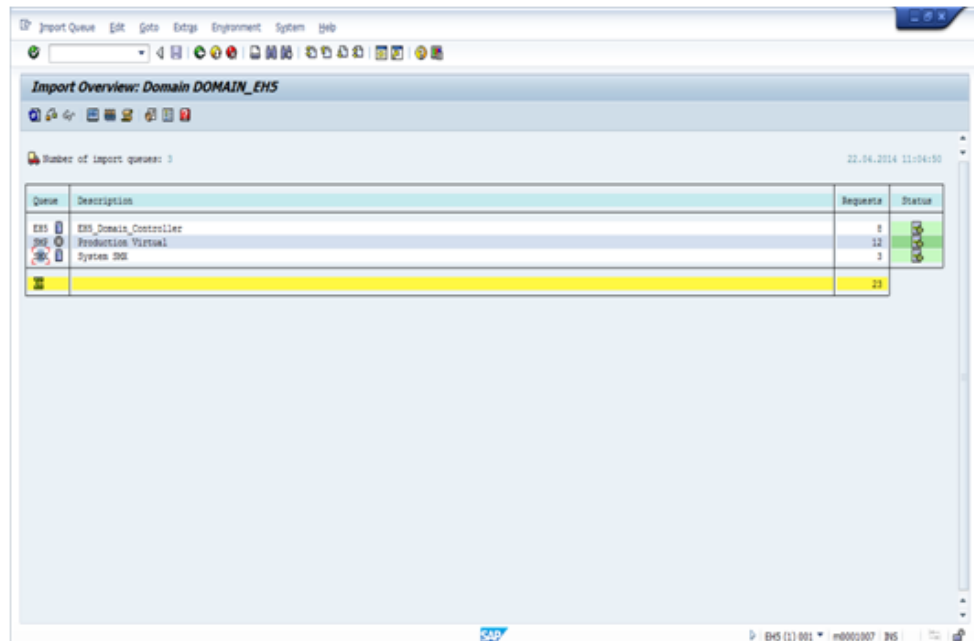
Perform the following steps to install the TR files:

1. Log in to the desired SAP System where you want to import the Transport Request.
2. In the command field, type Tcode `STMS` and press **Enter**.



The Transport Management System screen appears.

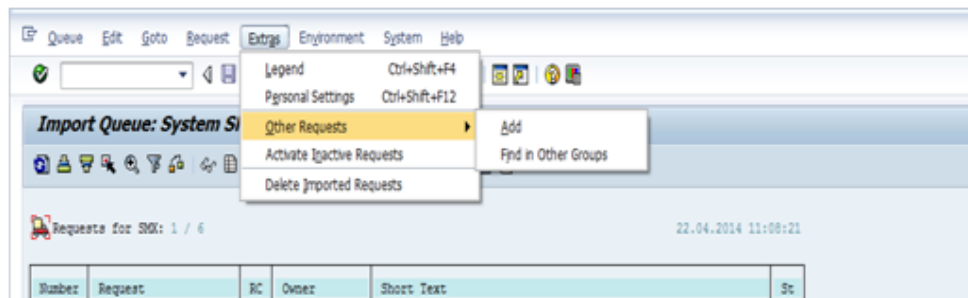
3. Click the  icon or press **F5**.
4. In the Import Overview screen, double-click the target system. In this case, transport request is imported to Sandbox server by double clicking on SMX.



5. Refresh to update the transport list.

Number	Request	RC	Owner	Short Text	St
1	S10K900959		RBANSAL		
2	SAPK-70211INCTS9PLGN		SAPUSER	CTS service plugin 702 version 11 (20130502)	
3	SAPK-57212INCTS9PLGN		SAPUSER	CTS server plugin 702 version 12 (20130704)	
4	U1SK902328		SAP	Potential modif. of persisted data in RFC_RSUPO_EXEC	
5	P1ZKD01341		GOETZMA	Transport of copies for security fix MW	
6	Y7SKD00276		SAP	Delete EHFI specific artifacts	

- If you want to add other Transport Request, click **Extras> Other Requests> Add** on the menu.



The added Transport Request can be seen in the System Import Queue.

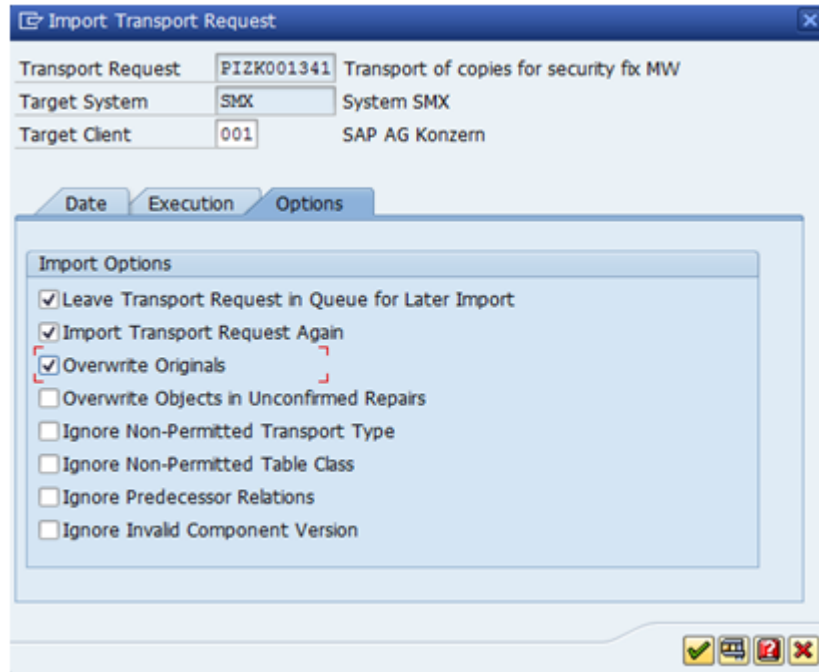
- Type the Transport Request Number, click **Import Again** check box, and then click **Continue**.

Transp. Request: S10K900959

Import Queue: SMX System SMX

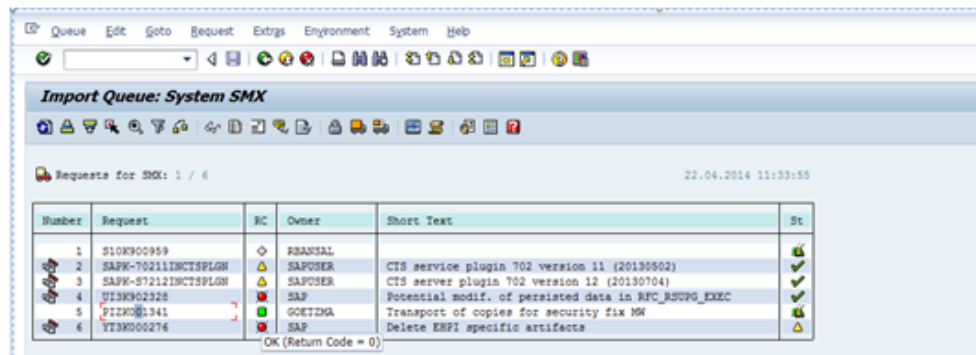
Import Again:

- Select the Transport Request and click the icon or press **Ctrl + F11**.
- Specify your target client, for example, Client is 001.
- Click the **Options** tab, choose first three options and click **Continue**.



11. On the Start Import dialog, click **Yes**.

The Transport Request import will start. Wait until the Transport Request import is completed successfully as shown below.



▲ Caution:

Artefacts of TR Files: Naming convention used is R<6 digits>.<source system> and K<6 digits>.<source system>.

The Source system is the developers source system.

K Type Transport : Cofile (1 - 3 KB in size)

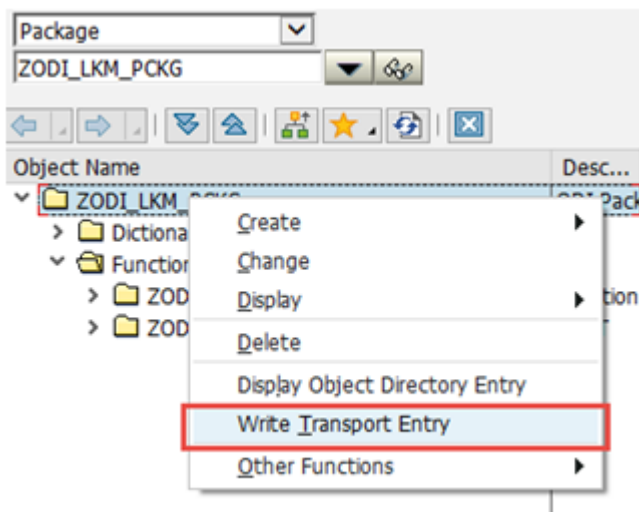
R Type Transport : Datafile (size more than the Cofile)

- **K file is a Cofile:** All transport requests' control and metadata attributes gets stored in this file, which are in the format of R<T.RNo>.<SID>. These are command or change request information files that include information about the transport type, object classes, required import steps, and post-processing exit codes.
- **R file is a Datafile:** All transport requests' data gets stored in this file, which are in the format of K<T.RNo>.<SID>. This contains the actual data for the Transport i.e. what changes will be made in your system.

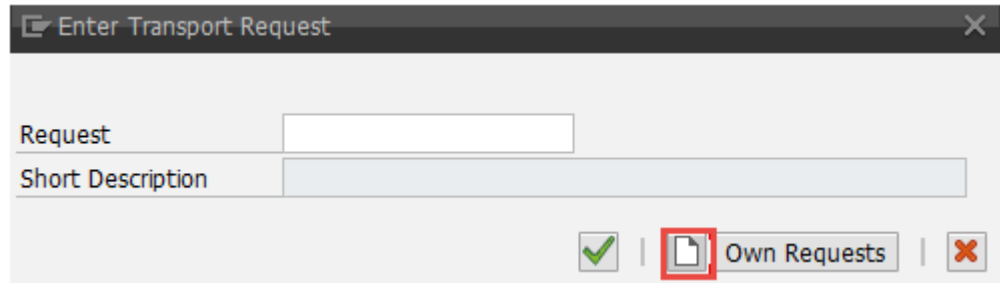
Creating Transport Request for ODI SAP Production Objects

Log in to the SAP system and perform the following steps to create transport request for ODI SAP Production Objects.

1. Go to Tcode SE80.
2. Provide package name ZODI_LKM_PCKG and press enter key.
3. Right-click the package name and click **Write Transport Entry**.



4. Click **All Objects** on the Transport Package pop-up message.
5. Click the **New Transport Request** button.



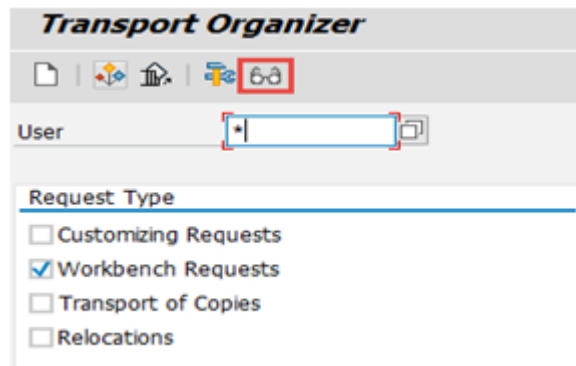
6. On the Select Request Type dialog, select **Workbench Request** and click **OK**.
7. Enter a description for the transport request and a valid target system and then click **Save**.

Target system name will be system default or as per configuration

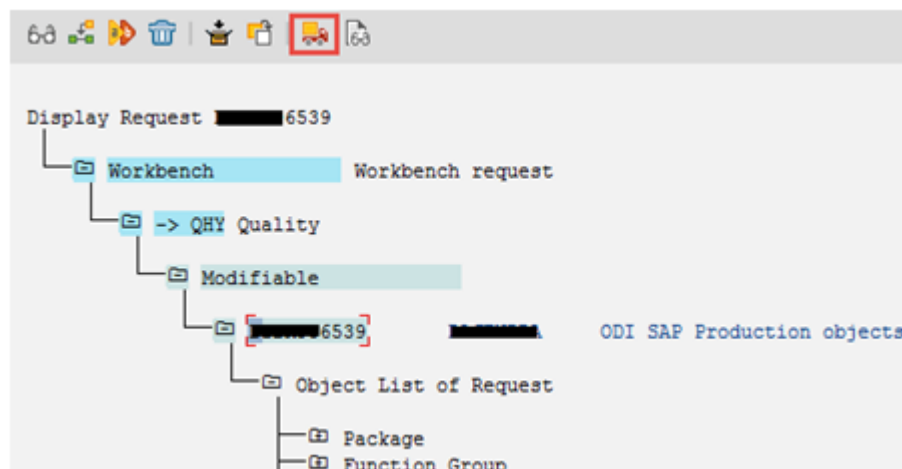
 **Note:**

Description should be ODI SAP Production Objects in order to easily identify and understand the grouping of SAP Objects.

8. Go to SE10 and click the **Display** button (shown in the following image).



9. On the Display dialog, enter the transport request number and click **OK**.
10. Expand the transport request tree and verify whether all the objects are included.

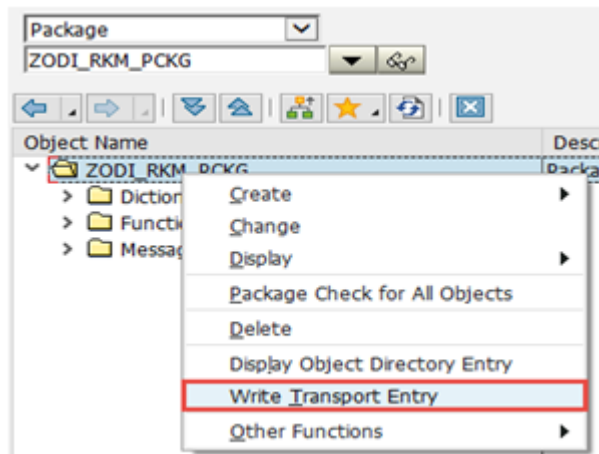


11. Copy the request number for further references.

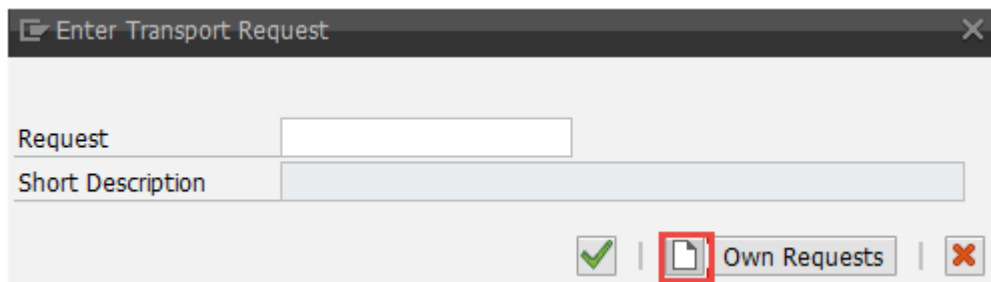
Creating Transport Request for ODI SAP Development Objects

Log in to the SAP system and perform the following steps to create transport request for ODI SAP Development Objects.

1. Go to Tcode SE80.
2. Provide package name ZODI_RKM_PCKG and press enter key.
3. Right-click the package name and click **Write Transport Entry**.



4. Click **All Objects** on the Transport Package pop-up message.
5. Click the **New Transport Request** button.



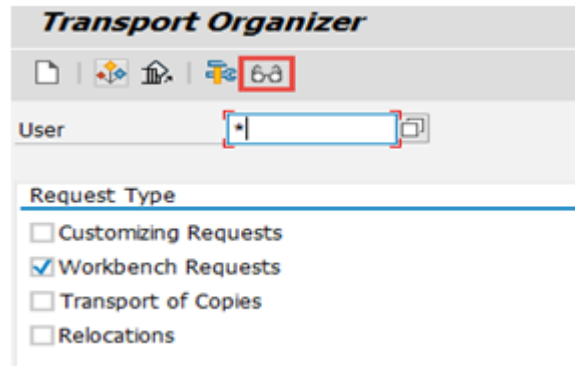
6. On the Select Request Type dialog, select **Workbench request** and click **OK**.
7. Enter a description for the transport request and a valid target system and then click **Save**.

Target system name will be system default or as per configuration

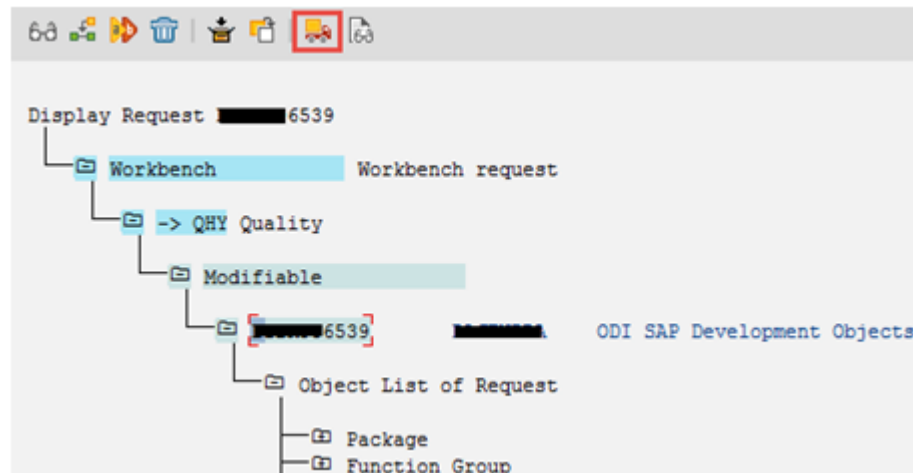
Note:

Description should be ODI SAP Development Objects in order to easily identify and understand the grouping of SAP Objects.

8. Go to SE10 and click the **Display** button (shown in the following image).



9. On the Display dialog, enter the transport request number and click **OK**.
10. Expand the transport request tree and verify whether all the objects are included.

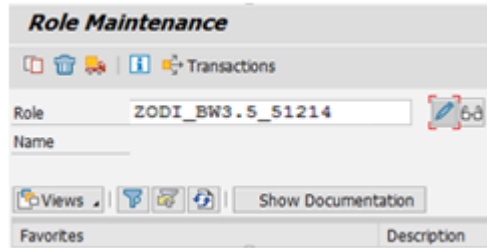


11. Copy the request number for further references.

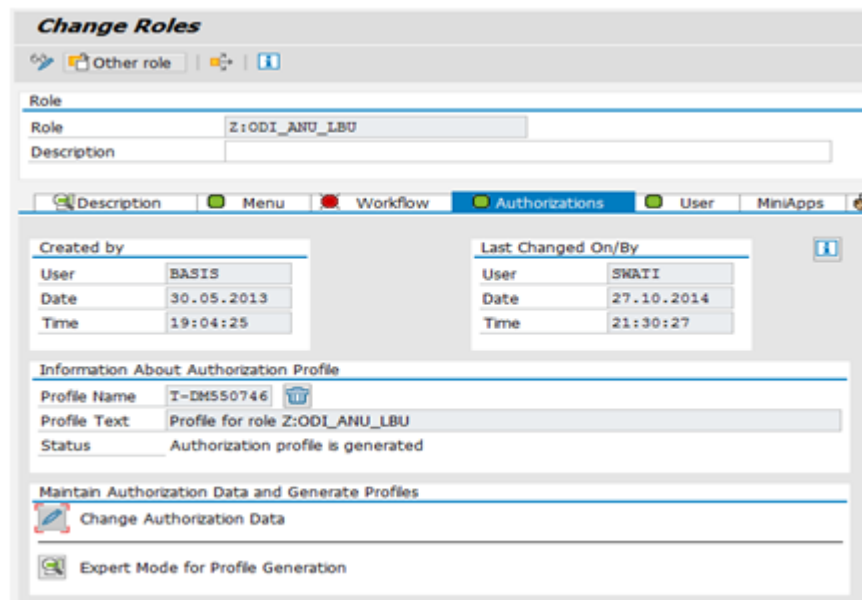
Installing and Assigning SAP User Profile

Perform the following steps to install and assign SAP user profile:

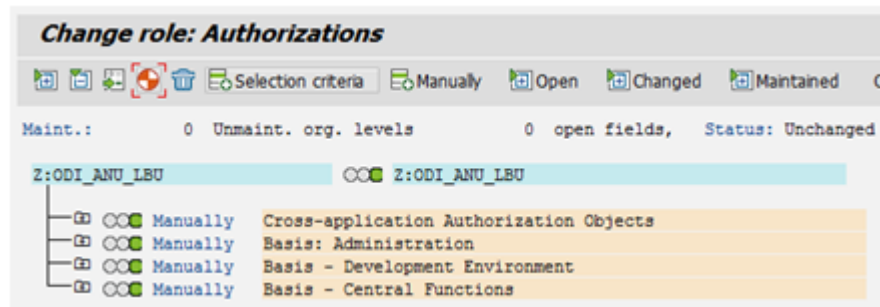
1. Execute PFCG, on the menu, click **Role** and then select **Upload**.
2. Click **Continue** on the Information dialog.
3. Specify the path where you have saved the .SAP file and then click **Open**.
4. Scroll down the drop down list and provide the appropriate input and click **Allow**.
5. Click **Continue** on the subsequent screen.
Role is successfully uploaded.
6. Type the name of uploaded role and click the **Edit** icon as shown below.



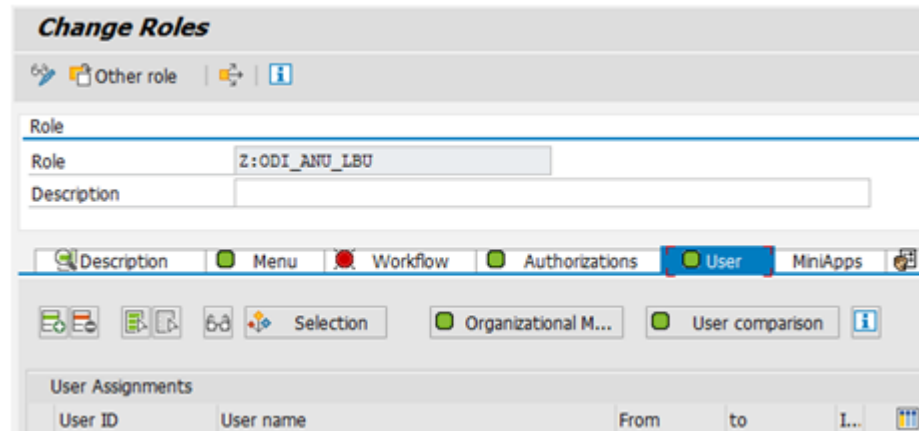
7. Click the **Roles** tab.
8. Perform the following steps to activate each role:
 - a. Double click the role Z:ODI_ANU_LBU.
 - b. Click the **Authorizations** tab.
 - c. Click the **Change Authorization Data** icon shown below.



- d. Click the **Generate**  icon.



- e. Click **Generate** and then click **Back**.
- f. Click the **User** tab and compare the user.



 **Note:**

The status of Authorizations and User tab must be green.

- g. Repeat these steps to activate for all roles inside Z:ODI_COMPOSITE.
9. Click **Save**.
10. Once the import process is done successfully, assign the imported profile to the SAP ODI user.
11. Request and assign SAP Developer License Key to ODI SAP User as described in section [SAP ABAP BW Required Privileges](#) or [SAP ABAP ERP Required Privileges](#). This is mandatory for the ODI SAP Adapter to work.

 **Note:**

In a few use-cases (use in production) the developer key can be omitted. For details refer to [SAP ABAP BW Required Privileges](#) or [SAP ABAP ERP Required Privileges](#).

Validating the ODI SAP Setup

This section describes some basic validation steps for the SAP administrators. Executing these steps ensures that previous installation steps were successful.

This section includes the following topics:

- [Validating the Shared Folder Setup](#)
- [Validating the FTP Setup](#)
- [Validating SAP Privileges](#)
- [Validating SAP Transport Layer Name](#)

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 ensure that you do NOT need to enter any credentials. If this does not work or you had to enter credentials, please contact your windows administrator or the system administrator of the system your shared folder is physically located on. Do not continue until this test passes.

3. Print screen.

Validating the FTP Setup

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

Step 1: Validating SAPFTP destination

1. Start the SAPGUI.

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

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

7. Print screen.

Step 2: Testing FTP connection

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

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

```
250 CWD successful.
```

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

9. Print screen.

Validating SAP Privileges

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

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

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

If a transaction allows the creation of a program without asking for any key or other authorization message, then the SAP user has validated that it has the appropriate dev rights and license key. Otherwise your SAP basis team needs to register the SAP

user in service.sap.com to get the license key and a Basis person can help him with dev rights.

Validating SAP Transport Layer Name

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

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

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

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

It is important to specify the correct transport layer name before running the RKM SAP ERP for the first time.

Perform the following steps to identify the list of defined transport layers in your SAP landscape:

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

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

Uninstalling ODI SAP Components

This section describes how to remove the ODI SAP Components from an SAP system. In case you plan to update to a newer ODI SAP Component version, see [Updating](#)

[ODI SAP Components](#) to identify whether the uninstallation step is required or can be skipped.

Perform the following steps to uninstall ODI SAP components:

1. Start the SAP GUI.
2. Connect to the SAP systems you want to uninstall.
3. Go to the transaction SE80 and provide Package name, for example, ZODI_LKM_PCKG.
4. Right-click the package name, and select **other options >rebuild object list** from the context menu.
5. Expand the object list and delete individual objects in the order listed below:
 - Program
 - Function group
 - Table type
 - Structure
 - Message class

If asked for transport request, create a new request and copy the requested number in Notepad.

6. Repeat step 4.

If any object is remaining, it will be displayed. Delete the object.
7. Right-click the package name, go to **display object directory entry> press lock overview** button, copy the request number, and save it in Notepad.
8. Go to transaction SE10, and release all the transport requests.
9. Go to Transaction SE16, enter package name in **DEVCLASS** field of table TADIR.

There should be a single record for the package name. If there are entries in the table, then go to SE11 and table E071. Specify the Object and object name whose entry is in the TADIR.
10. Mark the field **LOCKFLAG** as **X** and execute.

From there you will get the task or request. Go to Tcode SE01 and release this request.
11. Go to transaction SLG0, and search for zodi* select message class and go to sub-objects and delete the object.

Now press back button and delete message class. Press **Save** button. It will ask for transport request, create a new request and save it in Notepad.
12. Delete the package from SE80.

It will ask for a transport request. Create the request and release it using transaction SE10.
13. Repeat the steps 3 to 12 for package ZODI_RKM_PACKAGE.

If any of the objects is not deleted, then go to SU53 and check missing privileges. Try to grant the same to that SAP user and then continue with deleting that object.

See [Validating Uninstallation of ODI SAP Components](#) for information on how to validate uninstallation of ODI SAP components.

Validating Uninstallation of ODI SAP Components

Perform the following steps to validate the uninstallation of the ODI SAP components:

1. Go to transaction SE11.
2. Enter the table name TADIR and click **Display**.
3. On the Application toolbar, click **Contents**.
4. Specify the package name in the field **DEVCLASS** and press **F8**.

If there are no entries, a message **no table entries found** is shown.

If any entries are listed, the uninstallation is incomplete and reinstallation must not be attempted.

5. Go to transaction SE11. Enter the table name T100 and click **Display**.
6. On the Application toolbar, click **Contents**.
7. Type `zodi*` in the field **ARBGB** and press **F8**.

If there are no entries, a message **no table entries found** is shown.

If any entries are listed, the uninstallation is incomplete and reinstallation must not be attempted.

E

Moving ODI and SAP Components from Development to Production

It is possible to move ODI and SAP Components from the Development environment to the Production environment.

In this appendix, system ECC 5.0 is used.

This appendix includes the following sections:

- [Transport Request \(TR\)](#)
- [Remote Function Call \(RFC\)](#)
- [Adding Mappings Under Same Transport Request](#)
- [Creating ODI Package](#)
- [Generating ODI Scenario](#)
- [Moving ODI Scenario to production repository \(optional\)](#)
- [Releasing ODI SAP Components TR into Production](#)
- [Releasing the ODI Mapping TR into Production](#)
- [Executing the ODI Scenario](#)

Transport Request (TR)

The ODI SAP Adapter uses different sets of SAP objects, which are bundled in TRs:

- **ODI SAP Adapter product TRs:** There are two product TRs, which are available for download and are installed during the ODI SAP Adapter product installation:
 - TR containing SAP objects relevant for **development**, for example, to allow ODI to retrieve metadata from SAP system.
 - TR containing SAP objects relevant for **production**, for example, to run and monitor data extraction jobs.
- **ODI generated TRs:** During development the ODI SAP Adapter generates ABAP programs. These programs can be (automatically) uploaded and will be part of TRs. Depending on users choice there can be one or several TRs containing ABAP programs for data extraction.

This sections describes these TRs and how they can be inspected in SAP.

 **Note:**

Please note the following:

- TR is released by SAP BASIS team.
- There is a possibility that TR may automatically move to Production. This happens if Transport Route is configured. In most cases, the Transport Route to Production from Development or QA is not configured, since manually importing the TR is considered better. In case of an automatically configured TR, emphasis should be on correct sequence of TR, since the sequence plays an important role.
- In a distributed Production environment, the workbench TR should be imported in all of the application servers.

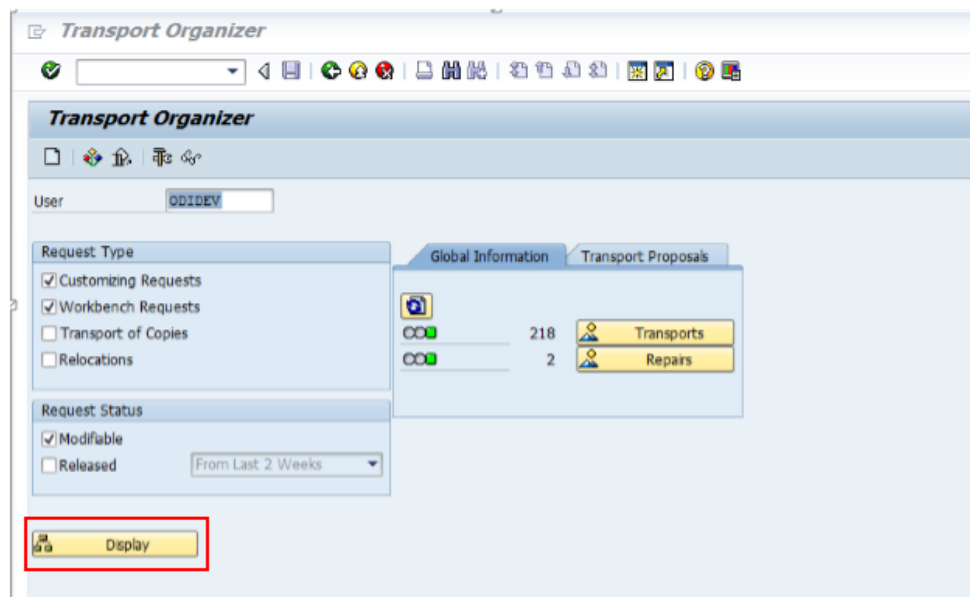
This section includes the following topics:

- [Viewing the List of TRs](#)
- [Transport Request \(TR\) relevant to DEV](#)
- [Transport Request \(TR\) relevant to PROD](#)

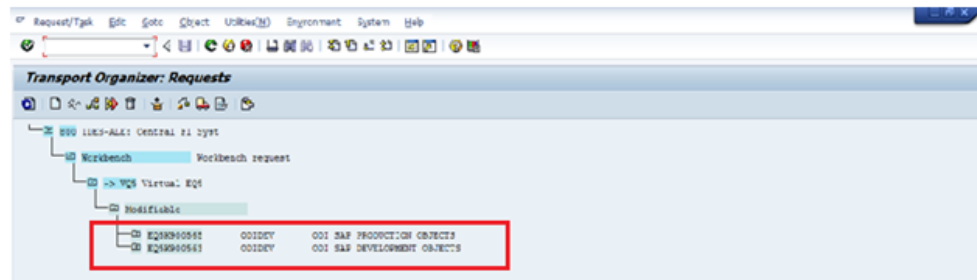
Viewing the List of TRs

Perform the following steps to view the list of TRs:

1. Type TCODE SE10.
2. Click the **Display** button that is located toward the bottom left corner of the screen.



Under Workbench, different target systems are listed. For example, VQ5, as shown below.



3. Expand the target system (example, VQ5), associated with the transport layer name (example, SAP) that was used at the time of RKM installation. The following two Transport Requests as shown above are displayed:
 - ODI SAP PRODUCTION OBJECTS
 - ODI SAP DEVELOPMENT OBJECTS

See [Description of TR Fields](#) for the description of TR fields.

Description of TR Fields

The following list describes the different TR fields.

- **TR Name:** EQ5K900562

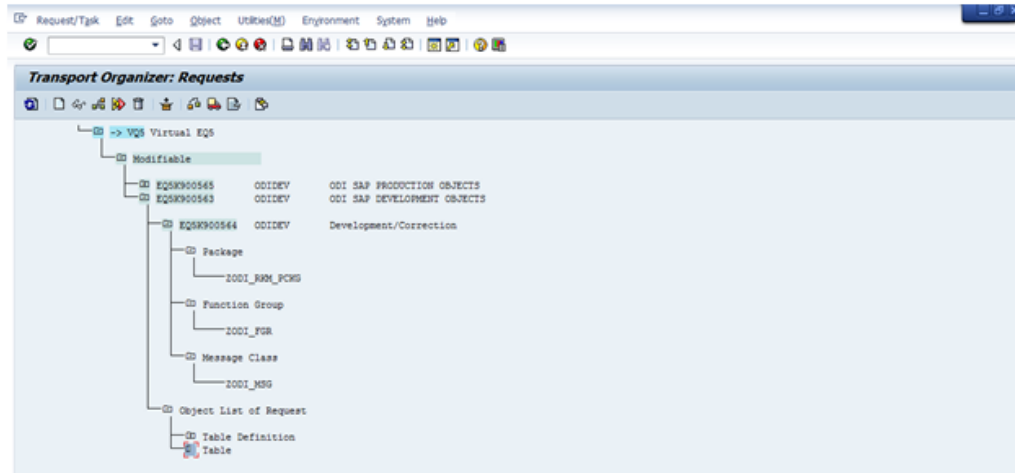
Note:

EQ5K900562, where EQ5 is the system identification (SID), K is a keyword, and the number is automatically generated from a range by the system, which starts at 900001 and is not to be maintained by the system administrators.

- **TR Description:** ODI SAP PRODUCTION OBJECTS
- **Transport Layer Name:** SAP

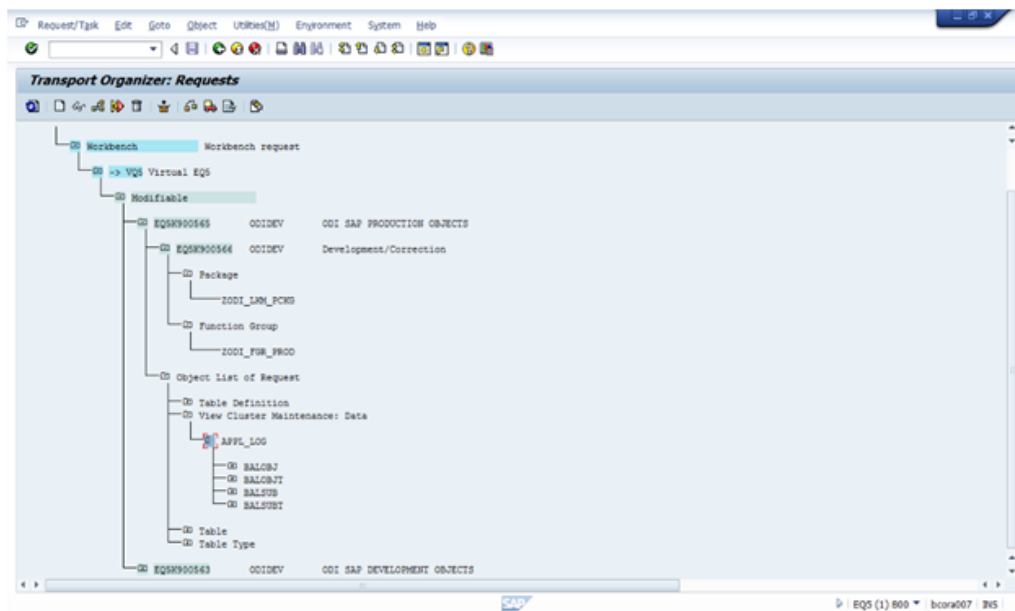
Transport Request (TR) relevant to DEV

The TR with description ODI SAP DEVELOPMENT OBJECTS is relevant to Development. Expanding this will display **Package**, **Function Group**, and **Message Class** that are created under this TR.



Transport Request (TR) relevant to PROD

The TR with description ODI SAP PRODUCTION OBJECTS is relevant to Production. Expanding this will display **Package**, **Function Group**, **Object List of Request**, for example, APPL_LOG that are created under this TR.



Remote Function Call (RFC)

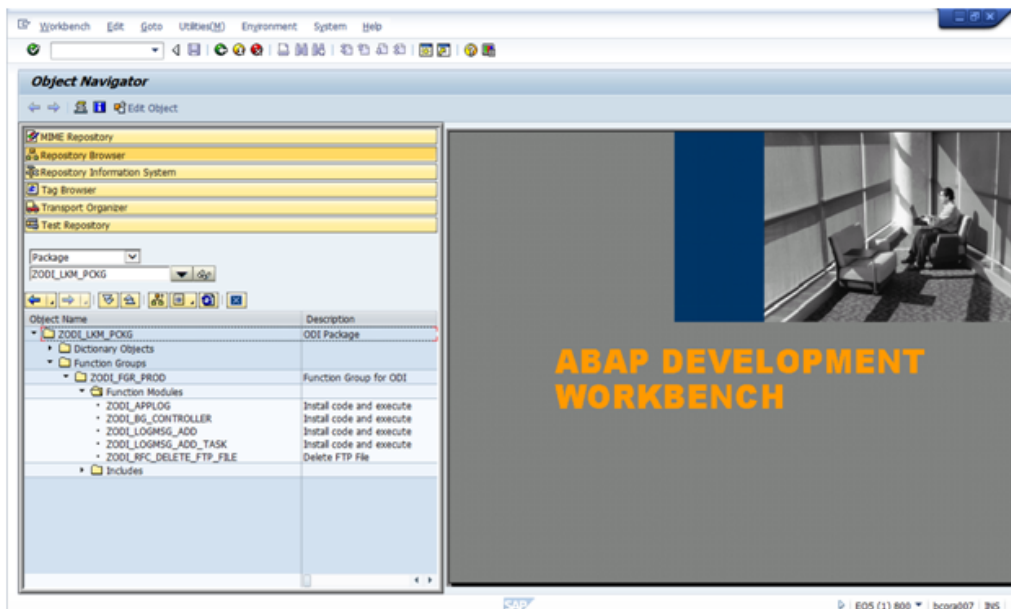
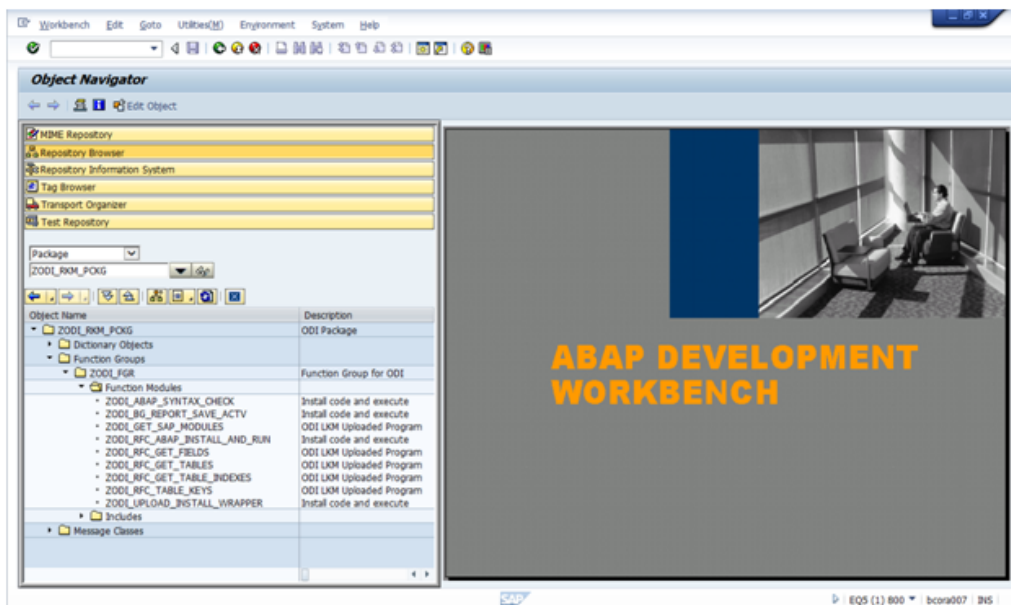
The entire set of generated RFCs are grouped under two SAP packages, based upon their execution environment. The following two SAP packages are created:

- **ZODI_RKM_PCKG**: contains all the RFCs required by the development SAP systems.
- **ZODI_LKM_PCKG**: contains all the RFCs required only by the production SAP system.

The list of RFCs created into SAP system can be seen by using TCODE SE80.

Perform the following steps to see a list of RFCs created in the SAP system:

1. Open TCODE SE80.
2. Below the option **Test Repository**, from the first drop-down list, select **Package**.
3. Type the package name (ZODI_RKM_PCKG/ZODI_LKM_PCKG) and press **Enter**.
4. Under **Object Name** segment, expand **Function Groups >ZODI_FGR >Function Modules**. A list of installed RFCs is displayed.



Adding Mappings Under Same Transport Request

By default the ODI SAP Adapter creates one TR per mapping. In case multiple mappings are used in a scenario, the user may want to move the generated ABAP programs inside SAP as a single unit. This is best achieved by grouping all ABAP programs into a single TR. This section describes how to let ODI group multiple ABAP programs into the same TR.

For background processing ODI combines all ABAP programs having the same `SAP_REPORT_NAME_PREFIX` into the same TR. The very first mapping executed defines the TR description in SAP via the KM option `SAP_TRANSPORT_REQUEST_DESC`. See [KM Options Descriptions](#) for more details.

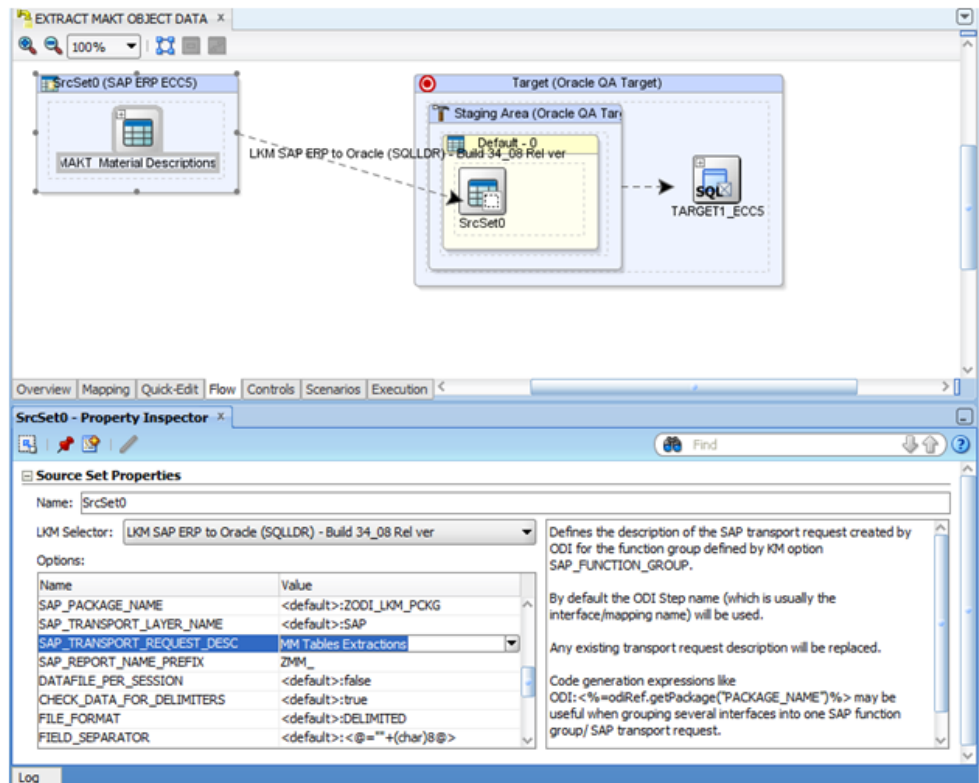
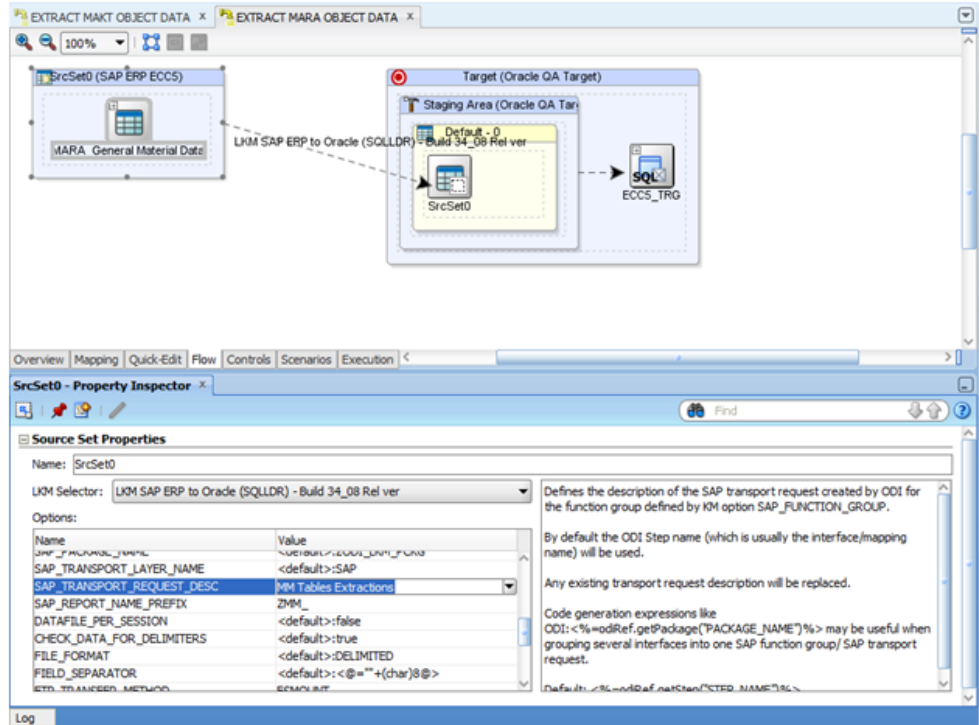
Perform the following steps to add interfaces under the same Transport Request:

1. Create two mappings. For example, `EXTRACT MAKT OBJECT DATA` and `EXTRACT MARA OBJECT DATA`.
2. In the Flow tab, set the value of option `BACKGROUND_PROCESSING` to `TRUE`.

Options:

Name	Value
<code>EXECUTE_ABAP_CODE</code>	<code><default>:true</code>
<code>ABAP_PROGRAM_NAME</code>	<code><default>:</code>
<code>BACKGROUND_PROCESSING</code>	<code><default>:true</code>
<code>JOB_CLASS</code>	<code><default>:A</code>
<code>LOG_FILE_NAME</code>	<code><default>:<?=new File((sBiAppsTempPath.equ...</code>
<code>TEMP_DIR</code>	<code><default>:<?=new File((sBiAppsTempPath.equ...</code>
<code>MAX_ALLOWED_ERRORS</code>	<code><default>:1</code>
<code>DELETE_TEMPORARY_OBJECTS</code>	<code><default>:true</code>

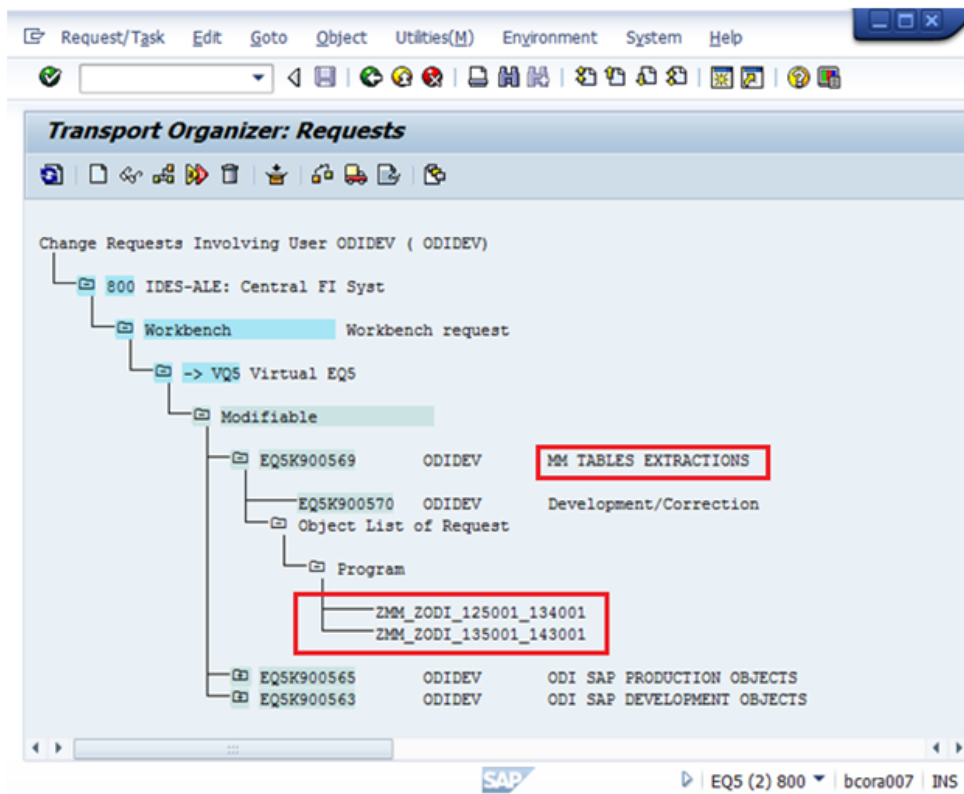
3. Enter the same `SAP_TRANSPORT_REQUEST_DESC`, for example, `MM Tables Extractions`, in both the mappings.
 For description of `SAP_TRANSPORT_REQUEST_DESC` option, see [KM Options Descriptions](#).
4. Enter `SAP_REPORT_NAME_PREFIX`, for example, `ZMM_`, in both the mappings.
 For description of `SAP_REPORT_NAME_PREFIX` option, see [KM Options Descriptions](#).



5. Execute both the mappings.
6. Go to SAP TCODE SE10.
7. Click the **Display** button.

The newly created transport request with the description provided in the KM option is displayed.

8. Expand the newly created **TR >Object List of Request > Program**, under this, the reports created by execution of LKM can be seen.



After following the above steps, both the mappings, ZMM_ZODI_125001_134001 and ZMM_ZODI_135001_143001 are added to the same transport request as shown in the image above.



Note:

In the Mapping ZMM_ZODI_135001_143001,
ZMM_: is the SAP_REPORT_NAME_PREFIX.
ZODI_135001_143001: dynamic ABAP program name.

The grouping of the above two ABAP programs is because of same value for SAP_REPORT_NAME_PREFIX KM option.

KM Options Descriptions

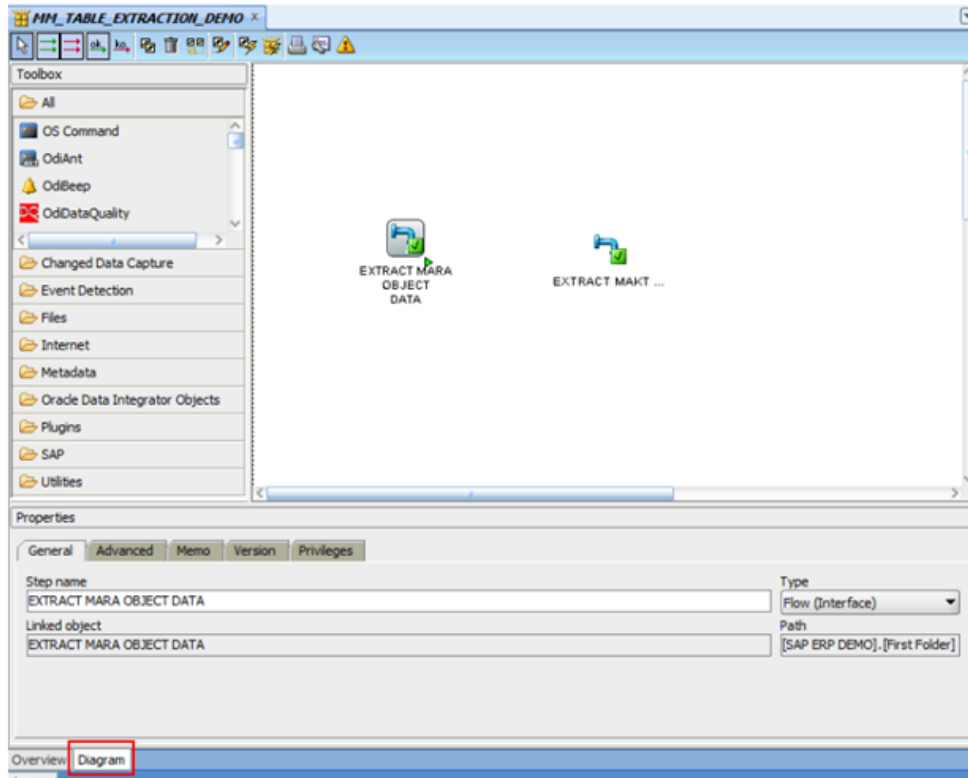
This section provides the description of the KM options SAP_REPORT_NAME_PREFIX and SAP_TRANSPORT_REQUEST_DESCRIPTION.

- **SAP_REPORT_NAME_PREFIX:** defines the prefix of all reports to be grouped into a common transport request, and is used in conjunction with `ABAP_PROGRAM_NAME`. Only applies if `BACKGROUND=1`. The prefix must start with Z. Please contact your SAP administrator for recommended report name prefixes.
- **SAP_TRANSPORT_REQUEST_DESCRIPTION:** defines the description of the SAP transport request created by ODI for the function group defined by KM option `SAP_FUNCTION_GROUP`. By default, the ODI Step name (which is usually the mapping name) will be used. Any existing transport request description will be replaced. Code generation expressions like `ODI:<%=odiRef.getPackage("PACKAGE_NAME")%>` may be useful when grouping several interfaces into one SAP function group/ SAP transport request.
Default: `<%=odiRef.getStep("STEP_NAME")%>`

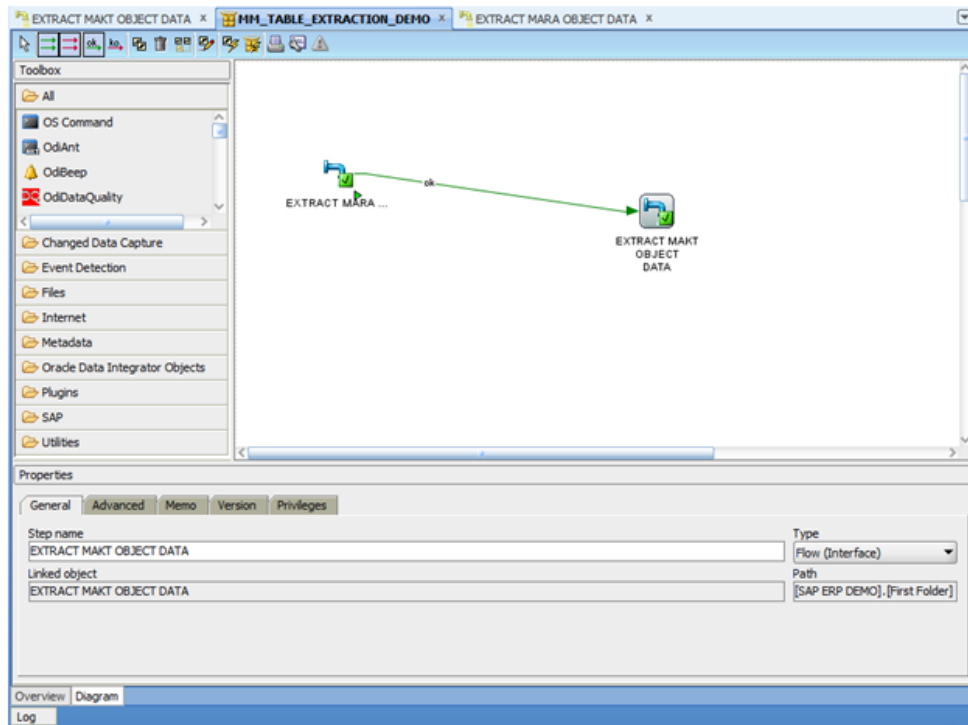
Creating ODI Package

Perform the following steps to create an ODI Package:

1. Go to SAP ERP project, for example, SAP ERP Demo, expand the **First Folder**, right-click **Packages** and select **New Package**.
2. In the Definition window, enter the name of the package, for example, `MM_TABLE_EXTRACTION_DEMO`, and click **Save**.
3. Click the **Diagram** tab.
4. To define the steps in the package, select `EXTRACT MARA OBJECT DATA` and `EXTRACT MAKT OBJECT DATA` interfaces from **Designer window->Project->First Folder -> Interfaces**, and drag-and-drop into the **Diagram** tab. These components appear as steps in the package and are not sequenced yet.
5. To specify the first step in the package, select and right-click this step, then select **First Step** from the context menu. A small green arrow appears on this step.



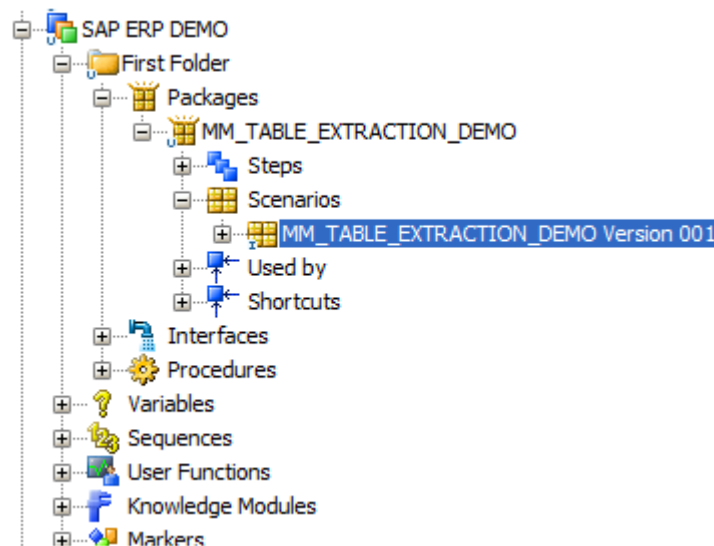
6. Select the **Next Step on Success** icon from the diagram tool bar.
7. To sequence the steps with this tool, click on one step, then click on the next step to be executed.



Generating ODI Scenario

Perform the following steps to generate an ODI Scenario:

1. Select the MM_TABLE_EXTRACTION_DEMO Package.
2. Right-click and select **Generate Scenario**.
The New Scenario dialog box appears.
3. Click **OK**. Oracle Data Integrator processes and generates the scenario.
The generated scenario is listed under Scenarios as shown below.



Moving ODI Scenario to production repository (optional)

This task is optional and only required if the scenario is to be executed from a different ODI repository.

Depending on the ODI Repository setup used, the generated ODI scenario must be moved into the production repository. For details on how to do this, refer *Exporting Scenarios* and *Importing Scenarios in Production* in *Developing Integration Projects with Oracle Data Integrator*.

In case the scenario will be executed from a different ODI software installation, ensure that all connectivity requirements have been met. See [Connectivity Requirements](#) or [Connectivity Requirements](#), particularly [Installing and Configuring JCo](#) or [Installing and Configuring JCo](#).

Releasing ODI SAP Components TR into Production





To release the transport request to production, the transport's request number associated with that production (ZODI_LKM_PCKG) package is required.

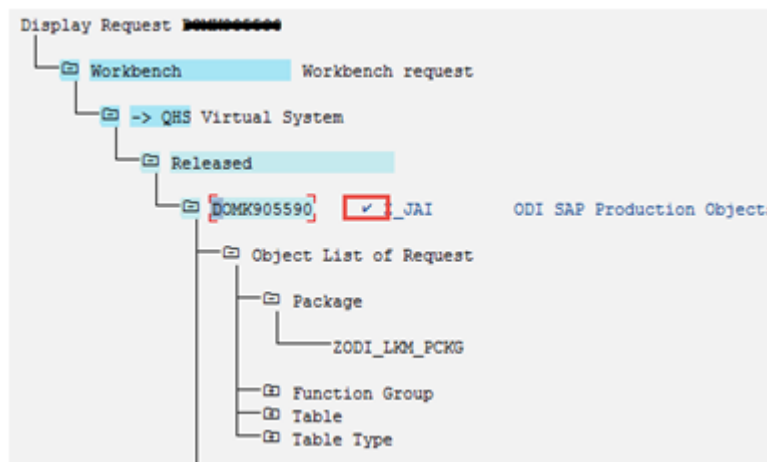
If the transports are created after the TR import, perform the steps mentioned in [Transports are created after TR import](#).

If the transports are created using the RKM process, perform the steps mentioned in [Transports are created using RKM process](#).

Transports are created after TR import

If transport requests are created after importing TR into development system and you have followed the instructions in section [Creating Transport Request for ODI SAP Production Objects](#) and [Creating Transport Request for ODI SAP Development Objects](#) carefully, then perform the following steps.

1. Go to transaction SE10 and specify the user name by which you have followed instructions in [Creating Transport Request for ODI SAP Production Objects](#) and [Creating Transport Request for ODI SAP Development Objects](#) or * for all users.
2. If you have noted the TR number, then click the **Display** button () and provide transport request number in subsequent popup.
3. On the Display dialog, enter the transport request number and click **OK**.
The transport request number with included objects appears.
4. If you don't remember the transport number then press display button () and search for ODI SAP Production Objects or the description specified in [Creating Transport Request for ODI SAP Production Objects](#).
This will take you to the corresponding request number.
5. Click **Position cursor** () button to locate the TR.
6. Selected transport is highlighted.
7. Select the transport request and click **Release** button () on the toolbar.



Note:

A check mark indicates that the transport request has been released.

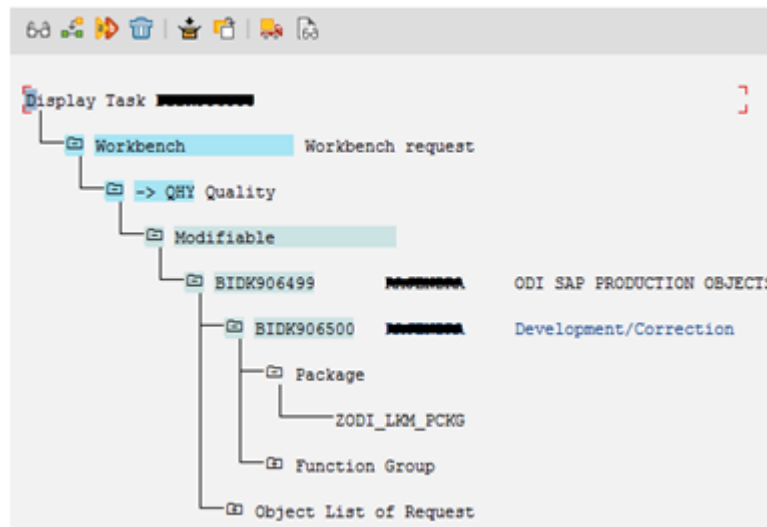
Transports are created using RKM process

If transport request are created using RKM process then perform the following steps.

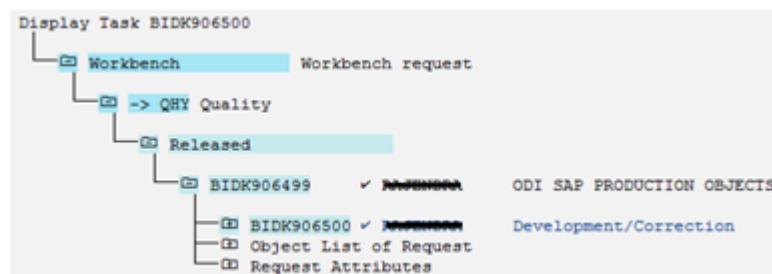
1. Go to TCODE SE16 of SAP system, and enter E071 in Table Name field.
2. Press Enter on the keyboard or click **View content** (📄) button.
3. Enter the package name ZODI_LKM_PCKG for **OBJ_NAME (Object Name)** and X for **LOCKFLAG (Status)** field.
4. Press F8 or execute.

This gives the transport request number. Copy the displayed transport request number.

5. Go to Transaction SE01 and open the TR copied as copied in the previous step, provide TR number and click the **Display** button.



6. Select task BIDK906500 (shown above) and click the **Release** button (🚚) and again select transport request number BIDK906499 (shown above) and click the **Release** button (🚚).



 **Note:**

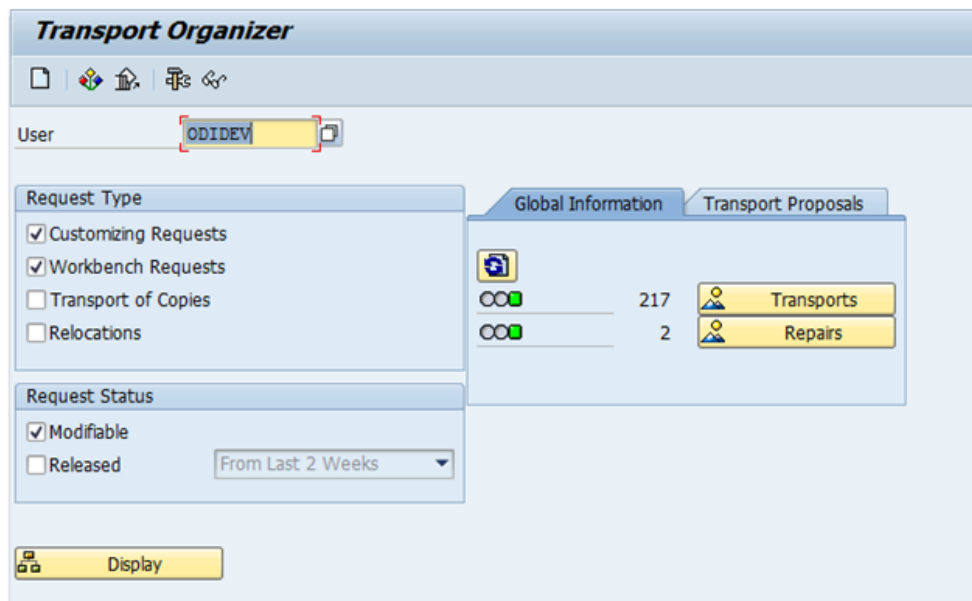
A check mark indicates that the transport request has been released.

Releasing the ODI Mapping TR into Production

To release the transport request of the mapping, the transport's request number associated with that mapping is required.

Perform the following steps to find the TR associated with the mapping:

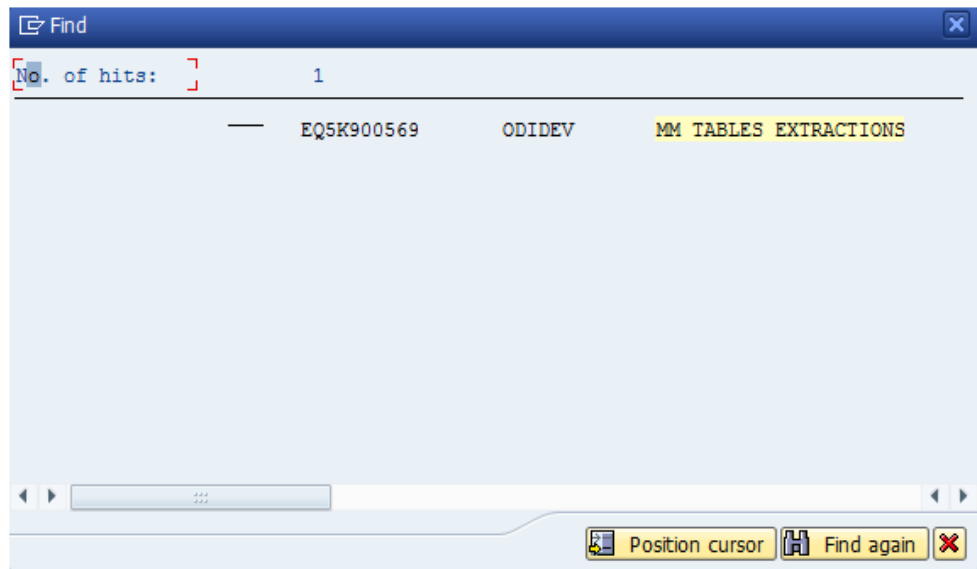
1. Go to TCODE SE10 in SAP system, and enter the user name used for executing the LKM.
2. Change the **Request Status** to **Modifiable** by checking it.



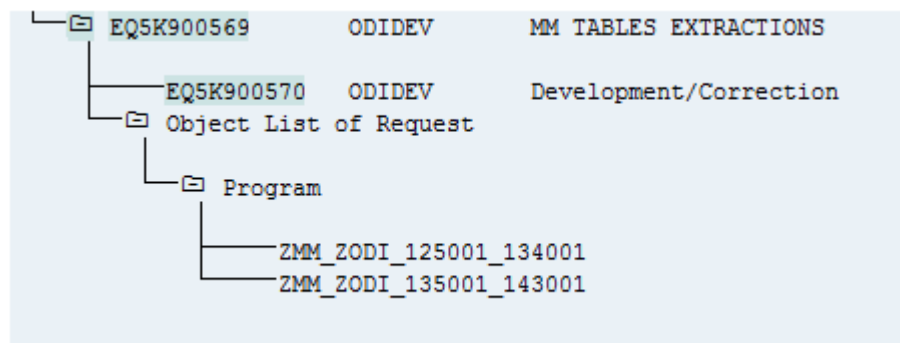
3. Click **Display**.
4. Press **CTRL+F** to search. Provide the SAP_TRANSPORT_REQUEST_DESC value which was entered in the interfaces while executing the LKM, for example, MM TABLES EXTRACTION.



- Click the Search icon.
A pop-up displaying the transport request number with its description appears.



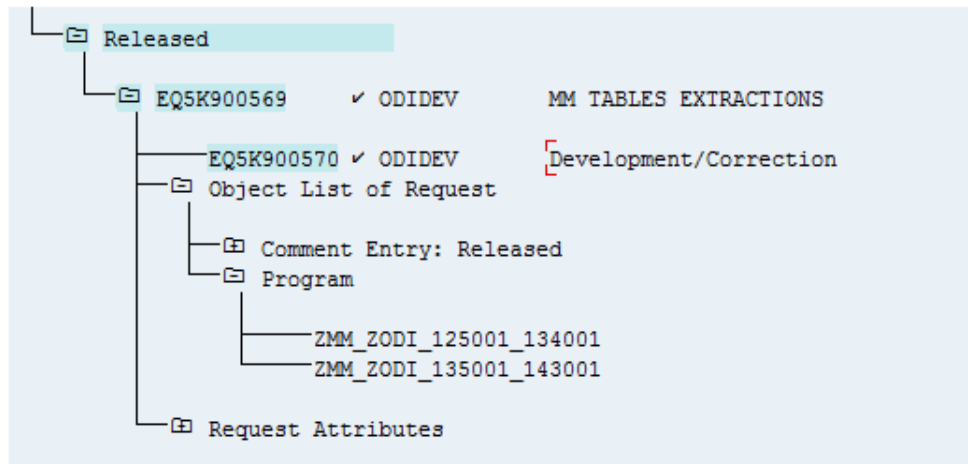
- Double-click the Transport Request number.
The request number and task number along with the objects under that request are shown.



7. Click the task number below the request number, and click the icon.
8. Click the request number, and click the icon.

 **Note:**

A check mark indicates that the Transport Request has been released.

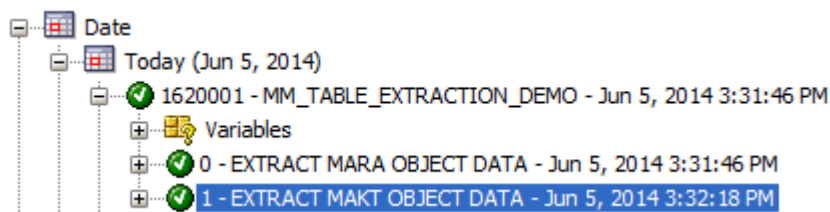


Executing the ODI Scenario

Perform the following steps to execute the ODI Scenario:

1. Right-click the scenario MM_TABLE_EXTRACTION_DEMO Version 001.
2. Select **Execute** from the context menu.
3. Click **OK** on the **Execution** and **Information** dialogs.

The scenario execution report can be reviewed in Operator, and the same results are seen as those obtained when the package was executed.



It is also possible to review the scenario execution report from the Designer module, by expanding the scenario tree view.

