

Oracle® Fusion Middleware

Application Adapters Guide for Oracle Data Integrator

12c (12.1.2)

E39360-03

January 2014

Copyright © 2010, 2014, Oracle and/or its affiliates. All rights reserved.

Primary Author: Laura Hofman Miquel

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

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

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

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

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

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

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

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	vii
Audience	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions	viii
1 Introduction	
1.1 Terminology	1-1
1.2 Using This Guide	1-2
2 Oracle E-Business Suite	
2.1 Introduction	2-1
2.1.1 Concepts	2-1
2.1.2 Knowledge Modules	2-2
2.2 Installation and Configuration	2-2
2.2.1 System Requirements and Certifications	2-2
2.2.2 Technology Specific Requirements	2-3
2.2.3 Connectivity Requirements	2-3
2.3 Setting Up the Topology	2-3
2.3.1 Create an Oracle Data Server	2-3
2.3.2 Create an Oracle Physical Schema	2-3
2.4 Setting up an Integration Project	2-4
2.5 Creating an Oracle Model and Reverse-Engineering E-Business Suite Tables	2-4
2.5.1 Create an Oracle Model	2-4
2.5.2 Reverse-Engineer E-Business Suite Tables	2-4
2.6 Designing a Mapping	2-6
2.6.1 Loading Data from E-Business Suite	2-6
2.6.2 Integrating Data in E-Business Suite through the Open Interface	2-6
2.6.2.1 Managing Group IDs	2-7
2.6.2.2 Executing an Open Interface Program	2-8
3 JD Edwards World	
3.1 Introduction	3-1
3.1.1 Concepts	3-1
3.1.2 Knowledge Modules	3-1

3.2	Installation and Configuration.....	3-2
3.2.1	System Requirements and Certifications	3-2
3.2.2	Technology Specific Requirements	3-2
3.2.3	Connectivity Requirements.....	3-3
3.3	Setting Up the Topology	3-4
3.3.1	Create a Data Server	3-4
3.3.2	Create a Physical Schema	3-4
3.4	Setting up an Integration Project.....	3-5
3.5	Creating and Reverse-Engineering a Model.....	3-5
3.5.1	Create a Model	3-5
3.5.2	Reverse-Engineer JDE Tables.....	3-5
3.6	Designing a Mapping	3-6
3.6.1	Loading Data from JDE.....	3-6
3.6.2	Integrating Data in JDE.....	3-7

4 SAP ABAP BW

4.1	Introduction	4-1
4.1.1	Concepts.....	4-1
4.1.2	Knowledge Modules	4-2
4.1.3	Overview of the SAP BW Integration Process.....	4-2
4.1.3.1	Reverse-Engineering Process.....	4-3
4.1.3.2	Integration Process	4-3
4.2	Installation and Configuration.....	4-3
4.2.1	System Requirements and Certifications	4-4
4.2.2	Technology Specific Requirements	4-4
4.2.3	Connectivity Requirements.....	4-5
4.2.3.1	Installing and Configuring JCo	4-5
4.2.3.2	Gathering SAP Connection Information.....	4-6
4.2.3.3	Gathering FTP Connection Information	4-6
4.2.3.4	Gathering Shared Directory Information.....	4-7
4.2.4	Getting the Right Privileges	4-7
4.2.5	Adding the Open Tool	4-7
4.3	Defining the Topology	4-8
4.3.1	Create the File Data Server	4-8
4.3.1.1	Create a File Data Server	4-8
4.3.1.2	Create the File Schema.....	4-9
4.3.2	Create the SAP ABAP Data Server.....	4-10
4.3.2.1	Create the SAP ABAP Data Server.....	4-10
4.3.2.2	Create the SAP ABAP Schema.....	4-11
4.4	Setting up the Project.....	4-11
4.5	Creating and Reverse-Engineering a Model.....	4-11
4.5.1	Creating a SAP BW Model	4-12
4.5.2	Reverse-Engineering a SAP BW Model.....	4-12
4.6	Designing a Mapping	4-13
4.7	Considerations for SAP BW Integration.....	4-13
4.7.1	File Transfer Configurations	4-13
4.7.1.1	Transfer using a Shared Directory (recommended).....	4-13

4.7.1.2	FTP based Transfer	4-15
4.7.2	Execution Modes.....	4-17
4.7.3	Controlling ABAP Uploading / ABAP code in production	4-18
4.7.4	Managing ODI SAP Transport Requests.....	4-19
4.7.5	SAP Packages and SAP Function Groups	4-20
4.7.6	Log Files	4-21
4.7.7	Limitation of the SAP BW Adapter.....	4-21

5 SAP ABAP ERP

5.1	Introduction	5-1
5.1.1	Concepts.....	5-1
5.1.2	Knowledge Modules	5-1
5.1.3	Overview of the SAP ABAP Integration Process	5-2
5.1.3.1	Reverse-Engineering Process.....	5-2
5.1.3.2	Integration Process	5-2
5.2	Installation and Configuration.....	5-3
5.2.1	System Requirements and Certification.....	5-3
5.2.2	Technology Specific Requirements	5-3
5.2.3	Connectivity Requirements.....	5-4
5.2.3.1	Installing and Configuring JCo	5-4
5.2.3.2	Gathering SAP Connection Information.....	5-5
5.2.3.3	Gathering FTP Connection Information	5-6
5.2.3.4	Gathering Shared Directory Information.....	5-6
5.2.4	Getting the Right Privileges	5-6
5.2.5	Adding the Open Tool	5-6
5.3	Defining the Topology	5-7
5.3.1	Create the File Data Server	5-7
5.3.1.1	Create a File Data Server	5-7
5.3.1.2	Create the File Schema.....	5-8
5.3.2	Create the SAP ABAP Data Server.....	5-9
5.3.2.1	Create the SAP ABAP Data Server	5-9
5.3.2.2	Create the SAP ABAP Schema.....	5-10
5.4	Setting up the Project.....	5-10
5.5	Creating and Reverse-Engineering a Model.....	5-11
5.5.1	Creating a SAP ERP Model	5-11
5.5.2	Reverse-Engineering a SAP ERP Model.....	5-11
5.6	Designing a Mapping	5-12
5.7	Considerations for SAP ERP Integration.....	5-13
5.7.1	File Transfer Configurations	5-13
5.7.1.1	Transfer using a Shared Directory (recommended).....	5-13
5.7.1.2	FTP based Transfer.....	5-15
5.7.2	Execution Modes.....	5-17
5.7.3	Controlling ABAP Uploading	5-17
5.7.4	Managing ODI SAP Transport Requests.....	5-18
5.7.5	SAP Packages and SAP Function Groups	5-19
5.7.6	Log Files	5-20
5.7.7	Limitations of the SAP ABAP Adapter	5-21

A Additional Information for SAP ABAP BW Adapter

A.1	SAP ABAP BW Required Privileges	A-1
A.1.1	Important points to consider.....	A-2
A.1.2	Authorizations Required for RKM SAP BW Upload	A-2
A.1.3	Authorizations Required for RKM SAP BW Execution	A-3
A.1.4	Authorizations Required for LKM SAP BW Upload (Background mode)	A-4
A.1.5	Authorizations Required for LKM SAP BW Execution (Background mode).....	A-5
A.1.6	Authorizations Required for LKM SAP BW Upload (Dialog mode).....	A-6
A.1.7	Authorizations Required for LKM SAP BW Execution (Dialog mode).....	A-7
A.2	SAP Stand-Alone Connection Test.....	A-8
A.2.1	SAP Stand-Alone Connection Test.....	A-8
A.3	Updating ODI SAP Components.....	A-9
A.4	Uninstalling ODI SAP Components.....	A-9

B Additional Information for SAP ABAP ERP Adapter

B.1	SAP ABAP ERP Required Privileges	B-1
B.1.1	Important points to consider.....	B-2
B.1.2	Authorizations Required for RKM SAP ERP Upload	B-2
B.1.3	Authorizations Required for RKM SAP ERP Execution	B-3
B.1.4	Authorizations Required for LKM SAP ERP Upload (Background mode)	B-3
B.1.5	Authorizations Required for LKM SAP ERP Execution (Background mode).....	B-4
B.1.6	Authorizations Required for LKM SAP ERP Upload (Dialog mode).....	B-5
B.1.7	Authorizations Required for LKM SAP ERP Execution (Dialog mode).....	B-5
B.2	SAP Connection Test	B-6
B.3	SAP Stand-Alone Connection Test.....	B-6
B.4	Updating ODI SAP Components.....	B-7
B.5	Uninstalling ODI SAP Components.....	B-8

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 have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following Oracle resources:

- *Installing and Configuring Oracle Data Integrator*
- *Upgrading Oracle Data Integrator*
- *Developer's Guide for Oracle Data Integrator*
- *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*
- *Knowledge Module Developer's Guide for Oracle Data Integrator*

- *Oracle Data Integrator 12c Online Help*
- *Release Notes for Oracle Data Integrator, included with your Oracle Data Integrator 12c installation and on Oracle Technology Network*

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.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Introduction

This chapter introduces the terminology used in this guide and provides information about how to use the guide.

Oracle Data Integrator uses Application Adapters to integrate data from and to enterprise applications.

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

- [Oracle E-Business Suite](#)
- [SAP ABAP BW](#)
- [SAP ABAP ERP](#)

This chapter includes the following sections:

- [Section 1.1, "Terminology"](#)
- [Section 1.2, "Using This Guide"](#)

1.1 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' 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.

1.2 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.

When these generic knowledge module can be used with a technology, the technology chapter will mention it. However, we recommend using 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.

Oracle E-Business Suite

This chapter describes how to work with Oracle E-Business Suite Knowledge Modules in Oracle Data Integrator.

This chapter includes the following sections:

- [Section 2.1, "Introduction"](#)
- [Section 2.2, "Installation and Configuration"](#)
- [Section 2.3, "Setting Up the Topology"](#)
- [Section 2.4, "Setting up an Integration Project"](#)
- [Section 2.5, "Creating an Oracle Model and Reverse-Engineering E-Business Suite Tables"](#)
- [Section 2.6, "Designing a Mapping"](#)

2.1 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.

2.1.1 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 a number of 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.

2.1.2 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 EBS Knowledge Modules

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>

2.2 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](#)

2.2.1 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/technology/products/oracle-data-integrator/index.html>.

2.2.2 Technology Specific Requirements

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

2.2.3 Connectivity Requirements

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

2.3 Setting Up the Topology

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

2.3.1 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.

2.3.2 Create an Oracle Physical Schema

Create an Oracle physical schema using the standard procedure, as described in "Creating a Physical Schema" of the *Developer's Guide for 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" of the *Developer's Guide for 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.

2.4 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 *Developer's Guide for 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.

2.5 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](#)

2.5.1 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 *Developer's Guide for 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.

2.5.2 Reverse-Engineer E-Business Suite Tables

The RKM E-Business Suite is able to 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 *Developer's Guide for 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 allows you to match any string of any length (including zero length)
 - _ the underscore symbol allows 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.

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.

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.

2.6 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.

2.6.1 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.

2.6.2 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 [Section 2.6.2.1, "Managing Group IDs"](#) and [Section 2.6.2.2, "Executing an Open Interface Program"](#).

2.6.2.1 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 you wish 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 you wish 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 you wish 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.

2.6.2.2 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, please 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

JD Edwards World

This chapter describes how to work with JD Edwards World Knowledge Modules in Oracle Data Integrator.

This chapter includes the following sections:

- [Section 3.1, "Introduction"](#)
- [Section 3.2, "Installation and Configuration"](#)
- [Section 3.3, "Setting Up the Topology"](#)
- [Section 3.4, "Setting up an Integration Project"](#)
- [Section 3.5, "Creating and Reverse-Engineering a Model"](#)
- [Section 3.6, "Designing a Mapping"](#)

3.1 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.

3.1.1 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)

3.1.2 Knowledge Modules

Oracle Data Integrator provides the Knowledge Modules listed in [Table 3-1](#) 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 3–1 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).
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

3.2 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](#)

3.2.1 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/technology/products/oracle-data-integrator/index.html>.

3.2.2 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 [Section 3.6.2, "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 3-2](#) 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.

3.2.3 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://195.10.10.13;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 Requirements" in the *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*.

3.3 Setting Up the Topology

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

3.3.1 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://10.139.142.183;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:
`jded://<host>;translatebinary=true;JDEWEnvironment=<name>;user=<name>;p
wd=<passwd>;JDEWTableNomenclature=OBJN_
OBJT;JDEWColumnNomenclature=FDFT_FDFN`

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

3.3.2 Create a Physical Schema

Create a physical schema under the data server that you have created in [Section 3.3.1, "Create a Data Server"](#). Use the standard procedure, as described in "Creating a Physical Schema" of the *Developer's Guide for 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" of the *Developer's Guide for Oracle Data Integrator* and associate it in a given context.

3.4 Setting up an Integration Project

Setting up a project using JDE World features follows the standard procedure. See "Creating an Integration Project" of the *Developer's Guide for 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*.

3.5 Creating and Reverse-Engineering a Model

This section contains the following topics:

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

3.5.1 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 *Developer's Guide for Oracle Data Integrator*.

3.5.2 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 *Developer's Guide for 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 Systems

Default 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 Systems

Default 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.

3.6 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.

3.6.1 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"

3.6.2 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, as well as 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:
 1. Set the JDE_INVOKE option to *Yes*.
 2. If you want to create a security file, set the connection related options as shown in [Table 3-2](#).

Table 3-2 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 3–2 (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 Section 3.2.2, "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
```

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

Table 3–3 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
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 programm.

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

Option	Value	Notes
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 (WRKSPPLF) command.

This chapter describes how to work with SAP BW Knowledge Modules in Oracle Data Integrator.

This chapter includes the following sections:

- [Section 4.1, "Introduction"](#)
- [Section 4.2, "Installation and Configuration"](#)
- [Section 4.3, "Defining the Topology"](#)
- [Section 4.4, "Setting up the Project"](#)
- [Section 4.5, "Creating and Reverse-Engineering a Model"](#)
- [Section 4.6, "Designing a Mapping"](#)
- [Section 4.7, "Considerations for SAP BW Integration"](#)

4.1 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 as well as step-by-step instructions including SAP connection testing.

4.1.1 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.

4.1.2 Knowledge Modules

Oracle Data Integrator provides the Knowledge Modules listed in [Table 4–1](#) 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 4–1 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 Appendix B, "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.

4.1.3 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.

4.1.3.1 Reverse-Engineering Process

Reverse-engineering uses the RKM SAP BW.

This knowledge module automatically installs dedicated RFC programs to retrieve SAP BW metadata. It 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.

4.1.3.2 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 [Section 4.7.1, "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.

4.2 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](#)
- [Getting the Right Privileges](#)

4.2.1 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/technology/products/oracle-data-integrator/index.html>.

4.2.2 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 [Section 4.2.1, "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 [Section 4.7.1, "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 of 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.

4.2.3 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](#)
- [Gathering SAP Connection Information](#)
- [Gathering FTP Connection Information](#)
- [Gathering Shared Directory Information](#)
- [Adding the Open Tool](#)

4.2.3.1 Installing and Configuring JCo

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

To install and configure JCo:

1. Download a supported JCo version for your configuration from <http://service.sap.com/connectors>. Check the supported JCo version in the Compatibility Matrix available at Oracle Technology Network: http://www.oracle.com/technology/products/oracle-data-integrator/10.1.3/htdocs/documentation/odi_certification.xls

Notes:

- 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.
 5. Restart the ODI Components using SAP (ODI Studio, Standalone Agent)
 6. Check the JCo installation.

4.2.3.2 Gathering SAP Connection Information

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

- SAP BW System IP Address or Hostname: IP address/ 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 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, 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.

4.2.3.3 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.

4.2.3.4 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.

4.2.4 Getting the Right Privileges

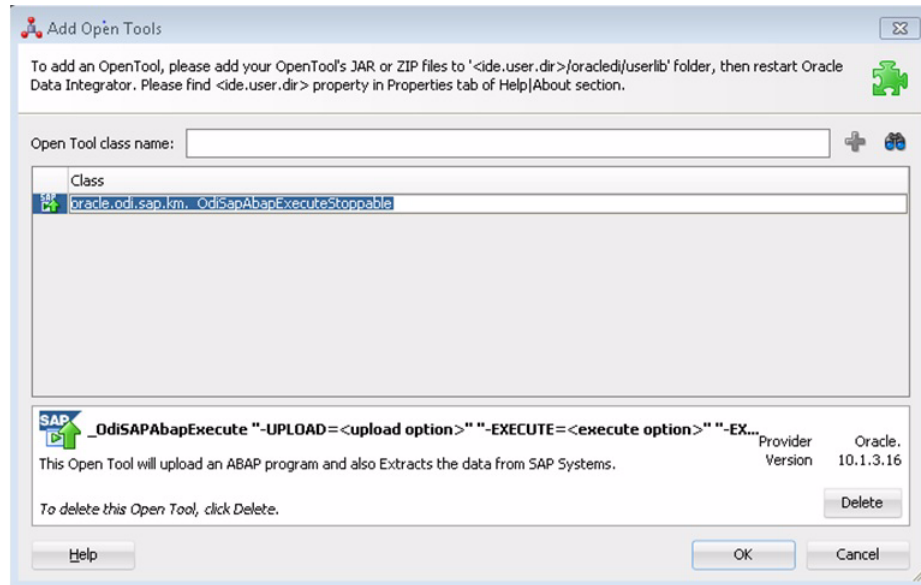
The SAP Adapter requires privileges to perform setup and execution operations. Please provide your administrators with the list of privileges listed in [Appendix A, "Additional Information for SAP ABAP BW Adapter"](#).

These privileges are required for the SAP user to log in the SAP System with ODI. This user is specified when creating the SAP data server.

4.2.5 Adding the Open Tool

The complete process of installing and adding an Open Tool to ODI is described in Appendix A, "Oracle Data Integrator Tools Reference" of *Developer's Guide for Oracle Data Integrator*. This section details only the SAP ABAP specific steps.

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



6. Click OK.

4.3 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](#)

4.3.1 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.

4.3.1.1 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, please see details on ENCODING parameter in section 3.3.1, "Creating a File Data Server" of *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*. The encoding chosen on this URL must match the code page used by the SAP Application Server.

See [Section 4.7.1, "File Transfer Configurations"](#) for more information.

4.3.1.2 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 (SQLDR)/ 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 (SQLLDR)/ 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 [Section 4.7.1, "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`.

4.3.2 Create the SAP ABAP Data Server

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

4.3.2.1 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 *Developer's Guide for Oracle Data Integrator*. This data server uses the SAP connection information.
2. Set the parameters for this data server as follows:
 - Name: `SAP_BW`. The name of the data server as it will appear in ODI.
 - Host (Data Server): SAP BW System IP Address or host name.
 - User: SAP BW User, as provided by the SAP Administrator.
 - Password: This user's SAP BW Password. This password is case-sensitive.
3. 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 Client Number: The three-digit number assigned to the self-contained unit which is called Client in SAP. A Client can be a training, development, testing or production client or represent different divisions in a large company.
 - SAP System Number: The two-digit number assigned to a SAP instance which is also called Web Application Server or WAS.
 - SAP System ID: The three-character, unique identifier of a SAP system in a landscape.
 - SAP SNC Connection Properties: SNC Connection Properties. This parameter is optional and can be left empty.
 - SAP Router String: Router String. This parameter is optional and can be left empty.
 - SAP Character Set: The character set is only required if your SAP system is not a UNICODE system. For a complete list of character sets, see "Locale Data" in the *Oracle Database Globalization Support Guide*. For example, `EE8ISO8859P2` for Croatian Data. For UNICODE systems, use `UTF8`.

- SAP 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 1 if ABAP code can be uploaded on this SAP System. Typically set to 0 for any non-development system. See [Section 4.7.3, "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 [Section 4.7.3, "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.

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.

4.3.2.2 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" of the *Developer's Guide for 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" of the *Developer's Guide for Oracle Data Integrator* in the appropriate context.

4.4 Setting up the Project

Setting up a project using SAP BW features follows the standard procedure. See "Creating an Integration Project" of the *Developer's Guide for 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.

4.5 Creating and Reverse-Engineering a Model

This section contains the following topics:

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

4.5.1 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 *Developer's Guide for Oracle Data Integrator*.

4.5.2 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 *Developer's Guide for 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. *For first time RKM use only:* Set SAP_TRANSPORT_LAYER_NAME to the name of the transport layer your SAP basis admin has provided you with.

Caution:

- A wrong or invalid transport layer name will cause serious delays during the installation process. Please use only the value provided by your SAP Basis Admin.
 - For first time RKM use only: If you are using a custom value for KM options SAP_FUNCTION_GROUP_ODI_DEV and SAP_FUNCTION_GROUP_ODI_PROD, these function groups must *not* exist prior the first execution of the UPLOAD_ABAP_CODE and UPLOAD_ABAP_BASE set to `true`.
 - For first time RKM use only: If you are using a custom value for KM options SAP_PACKAGE_NAME_ODI_DEV and SAP_PACKAGE_NAME_ODI_PROD, these packages must *not* exist prior the first execution of the UPLOAD_ABAP_CODE and UPLOAD_ABAP_BASE set to `true`.
-
-

4. *For first time RKM use only:* Set UPLOAD_ABAP_CODE and UPLOAD_ABAP_BASE to `Yes`.

After the first execution this value must be set back to `No`.

5. Save the model.
6. Click **Reverse-Engineer** in the Model Editor toolbar.
7. Set UPLOAD_ABAP_CODE and UPLOAD_ABAP_BASE back to `No`.
8. *For first time RKM use only:* Validate in Operator that the session is now in status *running*.

If session has failed, please validate settings. Do not move on until all installation steps have been completed successfully.

9. The Tree Metadata Browser appears after the session is started. Select the data store object(s) to reverse.
10. 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.

4.6 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).

4.7 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](#)

4.7.1 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](#)

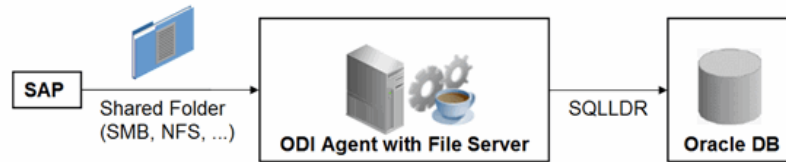
4.7.1.1 Transfer using a Shared Directory (recommended)

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 4–1 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.

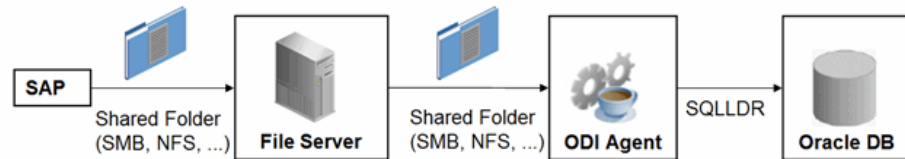
This configuration requires the following Topology settings:

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 4–2 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`.

4.7.1.2 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 4–3 Configuration 1: FTP Server is installed on an ODI Agent machine



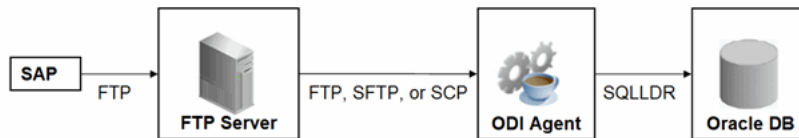
The configuration shown in Figure 4–3 is used, when `FTP_TRANSFER_METHOD = NONE`. In this configuration the following data movements are performed:

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 4–4 Configuration 2: FTP Server is not installed on ODI Agent machine



The configuration shown in Figure 4–4 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 (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 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.

4.7.2 Execution Modes

Background Processing

By default the generated ABAP code will be deployed as an ABAP report. At 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 backwards compatibility the KM option `BACKGROUND_PROCESSING` can be set to false. The generated ABAP code will then be deployed as an RFC. At execution this RFC is called by ODI to extract the data.

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

4.7.3 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, please make sure 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 Chapter 13, "Working with Scenarios" of *Developer's Guide for 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. By default, the name of the text file is similar to REPORT_ZODI_<Mapping Id>_<SourceSet Id>.ABAP. 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.

4.7.4 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. For valid function group names at your site please contact your SAP administrator.

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.

4.7.5 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 Adatper 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 [Section 4.7.4, "Managing ODI SAP Transport Requests"](#) for more information.

4.7.6 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 section "Runtime Logging for ODI components" of the *Developer's Guide for Oracle Data Integrator* for details about the ODI's logging facility. [Table 4–2](#) describes the different log files and their usage:

Table 4–2 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

4.7.7 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.
- 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'`

This chapter describes how to work with SAP ERP Knowledge Modules in Oracle Data Integrator.

This chapter includes the following sections:

- [Section 5.1, "Introduction"](#)
- [Section 5.2, "Installation and Configuration"](#)
- [Section 5.3, "Defining the Topology"](#)
- [Section 5.4, "Setting up the Project"](#)
- [Section 5.5, "Creating and Reverse-Engineering a Model"](#)
- [Section 5.6, "Designing a Mapping"](#)
- [Section 5.7, "Considerations for SAP ERP Integration"](#)

5.1 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 as well as step-by-step instructions including SAP connection testing.

5.1.1 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

5.1.2 Knowledge Modules

Oracle Data Integrator provides the Knowledge Modules listed in [Table 5–1](#) 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 5–1 SAP ERP Knowledge Modules

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 Appendix B.2, "SAP Connection Test" for more information.

5.1.3 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.

5.1.3.1 Reverse-Engineering Process

Reverse-engineering uses the RKM SAP ERP.

This KM automatically installs dedicated RFC programs to retrieve SAP metadata. It extracts the list of all SAP tables 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.

5.1.3.2 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 [Section 5.7.1, "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.

5.2 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](#)
- [Getting the Right Privileges](#)

5.2.1 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/technology/products/oracle-data-integrator/index.html>.

5.2.2 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 Technology Network (OTN). See [Section 5.2.1, "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 [Section 5.7.1, "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.

5.2.3 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](#)
- [Gathering SAP Connection Information](#)
- [Gathering FTP Connection Information](#)
- [Gathering Shared Directory Information](#)
- [Adding the Open Tool](#)

5.2.3.1 Installing and Configuring JCo

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

To install and configure JCo:

1. Download a supported JCo version for your configuration from <http://service.sap.com/connectors>. Check the supported JCo version in the Compatibility Matrix available at Oracle Technology Network:
http://www.oracle.com/technology/products/oracle-data-integrator/10.1.3/htdocs/documentation/odi_certification.xls

Notes:

- 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.
5. Restart the ODI Components using SAP (ODI Studio, Standalone Agent)
6. Check the JCo installation.

5.2.3.2 Gathering SAP Connection Information

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

- SAP ERP System IP Address or Hostname: IP address/ Hostname 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 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.

5.2.3.3 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.

5.2.3.4 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.

5.2.4 Getting the Right Privileges

The SAP Adapter requires privileges to perform setup and execution operations. Please provide your administrators with the list of privileges listed in [Appendix B, "Additional Information for SAP ABAP ERP Adapter"](#).

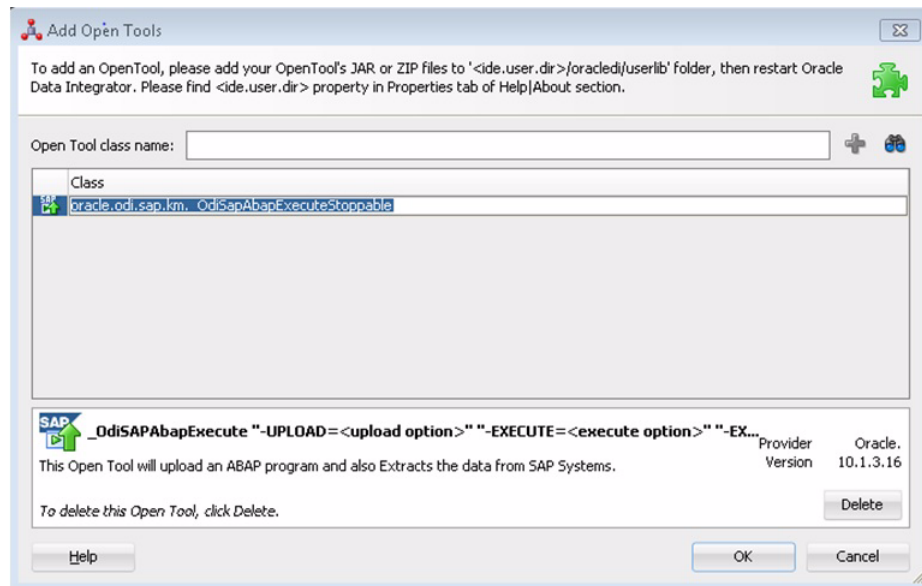
These privileges are required for the SAP user used to log in the SAP System with ODI. This user is specified when creating the SAP data server.

5.2.5 Adding the Open Tool

The complete process of installing and adding an Open Tool to ODI is described in Appendix A, "Oracle Data Integrator Tools Reference" of *Developer's Guide for Oracle Data Integrator*. This section details only the SAP ABAP specific steps.

1. Connect to Designer.
2. Select **File > Add/Remove Open Tools...**

3. In the Add/remove Open Tools window, enter the following name in the Open Tool class name field:
`oracle.odi.sap.km._OdiSapAbapExecuteStoppable`
4. Click **Add Open Tool**.
5. The Open Tool appears as shown below.



6. Click **OK**.

5.3 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](#)

5.3.1 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.

5.3.1.1 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, please see details on ENCODING parameter in section 3.3.1, "Creating a File Data Server" of *Connectivity and Knowledge Modules Guide for Oracle Data Integrator*. The encoding chosen on this URL must match the code page used by the SAP Application Server.

See [Section 5.7.1, "File Transfer Configurations"](#) for more information.

5.3.1.2 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 [Section 5.7.1, "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`.

5.3.2 Create the SAP ABAP Data Server

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

5.3.2.1 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 *Developer's Guide for Oracle Data Integrator*. This data server uses the SAP connection information.
2. Set the parameters for this data server as follows:
 - Name: The name of the data server as it will appear in ODI. For example, `SAP_ERP`.
 - Host (Data Server): SAP ERP System IP Address or Hostname
 - User: SAP User, as provided by the SAP Administrator
 - Password: This user's SAP Password. This password is case-sensitive.
3. 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 Client Number: The three-digit number assigned to the self-contained unit which is called Client in SAP. A Client can be a training, development, testing or production client or represent different divisions in a large company.
 - SAP System Number: The two-digit number assigned to a SAP instance which is also called Web Application Server or WAS.
 - SAP System ID: The three-character, unique identifier of a SAP system in a landscape.
 - SAP SNC Connection Properties: SNC Connection Properties. This parameter is optional and can be left empty.

- SAP Router String: Router String. This parameter is optional and can be left empty.
- SAP Character Set: 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.
- For a complete list of character sets, please refer to Oracle Database Globalization Support Guide: Locale Data: Character Sets. For example, EE8ISO8859P2 for Croatian Data."
- 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 [Section 5.7.3, "Controlling ABAP Uploading"](#) for more details.
- SAP Allow ABAP Execute: set to 1, if ABAP code can be executed on this SAP System. See [Section 5.7.3, "Controlling ABAP Uploading"](#) for more details.

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

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.

5.3.2.2 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" of the *Developer's Guide for 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" of the *Developer's Guide for Oracle Data Integrator* in the appropriate context.

5.4 Setting up the Project

Setting up a project using SAP ERP features follows the standard procedure. See "Creating an Integration Project" of the *Developer's Guide for 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.

5.5 Creating and Reverse-Engineering a Model

This section contains the following topics:

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

5.5.1 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 *Developer's Guide for Oracle Data Integrator*.

5.5.2 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 the *Developer's Guide for 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. *For first time RKM use only:* Set SAP_TRANSPORT_LAYER_NAME to the name of the transport layer your SAP basis admin has provided you with.

Caution: A wrong or invalid transport layer name will cause serious delays during the installation process. Please use only the value provided by your SAP Basis Admin.

4. *For first time RKM use only:* Set UPLOAD_ABAP_CODE and UPLOAD_ABAP_BASE to `Yes`.

After the first execution these values must be set back to `No`.

Caution:

- If you plan to install the ODI SAP ERP and the ODI SAP BW adapter into the **same** SAP system, please perform the installation of the ODI SAP BW adapter first. After successful installation of ODI SAP BW adapter, proceed here with the RKM options `ABAP_UPLOAD_BASE` set to `false` and `ABAP_UPLOAD_CODE` set to `true` (instead of both set to `true`). This ensures that the RKM SAP ERP will not try to install certain SAP base objects, which have already been installed by the RKM SAP BW.
 - For first time RKM use only: If you are using a custom value for KM options `SAP_FUNCTION_GROUP_ODI_DEV` and `SAP_FUNCTION_GROUP_ODI_PROD`, these function groups must *not* exist prior the first execution of the `UPLOAD_ABAP_CODE` and `UPLOAD_ABAP_BASE` set to `true`.
 - For first time RKM use only: If you are using a custom value for KM options `SAP_PACKAGE_NAME_ODI_DEV` and `SAP_PACKAGE_NAME_ODI_PROD`, these packages must *not* exist prior the first execution of the `UPLOAD_ABAP_CODE` and `UPLOAD_ABAP_BASE` set to `true`.
-

5. Save the model.
 6. Click **Reverse-Engineer** in the Model Editor toolbar.
 7. Set `UPLOAD_ABAP_CODE` and `UPLOAD_ABAP_BASE` back to `No`.
 8. *For first time RKM use only:* Validate in Operator that the session is now in status *running*.

If session has failed, please validate settings. Do not move on until all installation steps have been completed successfully.
 9. The Tree Metadata Browser appears after the session is started. Select the table(s) to reverse.
 10. 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.

5.6 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.

5.7 Considerations for SAP ERP Integration

This section includes the following topics:

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

5.7.1 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](#)

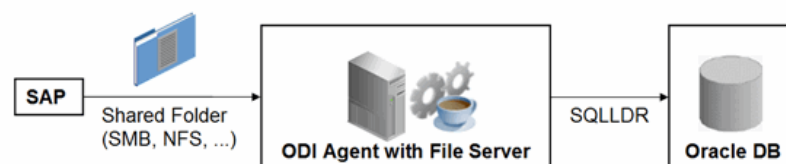
5.7.1.1 Transfer using a Shared Directory (recommended)

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 5–1 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.

- 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.

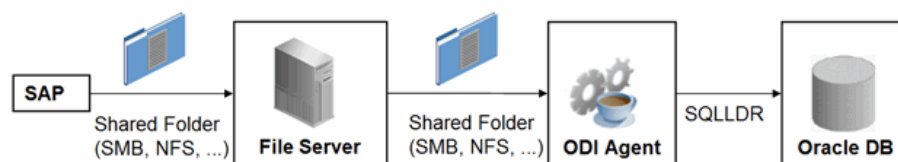
This configuration requires the following Topology settings:

- Create a File data server pointing to the File server/ODI Agent machine:
 - Host (Data Server): n/a
 - User: n/a
 - Password: n/a
- 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 5-2 Configuration 2



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

- The ABAP program extracts chunks of `FETCH_BATCH SIZE` records and writes them into a file in the shared folder.
- The run-time agent copies the file into the directory given by `TEMP_DIR` option of the LKM.
- 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 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.

5.7.1.2 FTP based Transfer

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 5–3 Configuration 1: FTP Server is installed on an ODI Agent machine



The configuration shown in [Figure 5–3](#) 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 (SQL*Loader): SQL*Loader reads locally the data file and loads the data into the Oracle staging area.

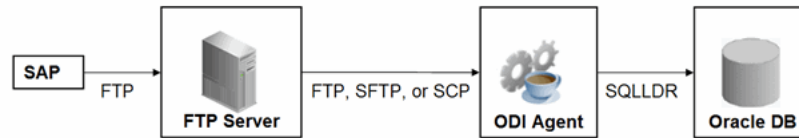
For LKM SAP ERP to SQL: The ODI File Driver reads locally the data file and inserts the data using JDBC into a non-Oracle staging area.

This configuration requires the following Topology settings:

1. Create a File data server pointing to the 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 5–4 Configuration 2: FTP Server is not installed on ODI Agent machine



The configuration shown in [Figure 5–4](#) 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.

5.7.2 Execution Modes

Background Processing

By default the generated ABAP code will be deployed as an ABAP report. At 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 backwards compatibility the KM option `BACKGROUND_PROCESSING` can be set to `false`. The generated ABAP code will then be deployed as an RFC. At execution this RFC is called by ODI to extract the data.

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

5.7.3 Controlling ABAP Uploading

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, please make sure 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 Chapter 13, "Working with Scenarios" of *Developer's Guide for 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. By default, the name of the text file is similar to `ZODI_<Mapping Id>_<SourceSet Id>.ABAP`. 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`.

5.7.4 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. For valid function group names at your site please contact your SAP administrator.

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.

5.7.5 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.

5.7.6 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 section "Runtime Logging for ODI components" of the *Developer's Guide for Oracle Data Integrator* for details about the ODI's logging facility. [Table 5-2](#) describes the different log files and their usage:

Table 5-2 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

Table 5–2 (Cont.) Log Files

Default Log File Name	KM / Phase	Content
<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

5.7.7 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 needs to be the one-table, for example when joining MARA and MARC, MARA needs to be the first table in the join.
- Mapping expression executed on the source must not contain any transformations.

Additional Information for SAP ABAP BW Adapter

This appendix describes the privileges that are required for connecting to SAP System, how you can test the connection outside of ODI using a standalone java utility, and how to uninstall SAP components.

This appendix includes the following sections:

- [Section A.1, "SAP ABAP BW Required Privileges"](#)
- [Section A.2, "SAP Stand-Alone Connection Test"](#)
- [Section A.3, "Updating ODI SAP Components"](#)
- [Section A.4, "Uninstalling ODI SAP Components"](#)

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

A.1 SAP ABAP BW Required Privileges

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

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

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

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

Table A-1 Required SAP User Types

Operation	Required SAP User Type
RKM: Setup	SAP Dialog user
RKM: Reverse Engineering	SAP Dialog user or SAP RFC user
LKM: Development	SAP Dialog user or SAP RFC user
LKM: Production	SAP Dialog user or SAP RFC user
Testing ODI Generated Extraction RFCs	SAP Dialog user or SAP RFC user

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

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

A.1.1 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**.

A.1.2 Authorizations Required for RKM SAP BW Upload

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

Note: Developer Key is required in this authorization.

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
S_RFC	ACTVT	16	
	RFC_NAME	RFC1, RS*, S*, SUTL, SYST, ZODI_FGR	
	RFC_TYPE	FUGR	
S_TCODE	TCD	SU53, SE38	
S_ADMI_FCD	S_ADMI_FCD		MENO
S_CTS_ADMI	CTS_ADMFCT	TABL	
S_DATASET	ACTVT	34	
	FILENAME	*	
	PROGRAM	SAPLSTRF	
S_TABU_DIS	ACTVT		03
	DICBERCLS		MA
S_DEVELOP	ACTVT	01, 02, 03, 06	07, 16
	DEVCLAS	\$TMP, ZODI_LKM_PCKG, ZODI_RKM_PCKG	

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
	OBJNAME	Z\$\$\$XRFC, ZODI_FGR, ZODI_FGR_PROD, ZODI_LKM_PCKG, ZODI_RKM_PCKG	
	OBJTYPE	DEVC, FUGR, PROG	
	P_GROUP		*
S_TRANSPRT	ACTVT	01,02,03,06	06 (NR)
	TTYPE	DTRA, TASK	
S_RS_ADMWB	ACTVT	03,16	16 (NR)
	RSADMWBOBJ	INFOOBJECT	
S_RS_ICUBE	ACTVT	03	
	RSICUBEOBJ	DATA, DEFINITION	DEFINITION
	RSINFOAREA	NODESNOTCONNECTED	0*
	RSINFOCUBE	0*, Z*	1*
S_RS_IOBJ	ACTVT	03	
	RSIOBJ	0*, Z*	
	RSIOBJCAT	*	0*, 0CHANOTASSINED, Z*
	RSIOBJPART	DEFINITION	DATA, DEFINITION
S_RS_ODSO	ACTVT	03	
	RSINFOAREA	0*, NODESNOTCONNECTED	
	RSODSOBJ	0*, Z*	
	RSODSPART	DATA, DEFINITION	

A.1.3 Authorizations Required for RKM SAP BW Execution

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

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
S_RFC	ACTVT	16	
	RFC_NAME	RFC1, RS*, S*, SUTL, SYST, ZODI_FGR	
	RFC_TYPE	FUGR	
S_TCODE	TCD	SU53	
S_TABU_DIS	ACTVT	03	
	DICBERCLS	MA	SC
S_DEVELOP	ACTVT	*	NR
	DEVCLASS	*	NR
	OBJNAME	*	NR

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
	OBJTYPE	*	NR
	P_GROUP	*	NR
S_RS_ADMWB	ACTVT	03, 16	
	RSADMWBOBJ	INFOBJECT	
S_RS_ICUBE	ACTVT	03	
	RSICUBEOBJ	DATA, DEFINITION	
	RSINFOAREA	NODESNOTCONNECTED	0*, NODESNOTCONNECTED (NR)
	RSINFOCUBE	0*, Z*	
S_RS_IOBJ	ACTVT	03	
	RSIOBJ	0*, Z*	Z* (NR)
	RSIOBJCAT	*	0*
	RSIOBJPART	DEFINITION	
S_RS_ODSO	ACTVT	03	
	RSINFOAREA	0*, NODESNOTCONNECTED	T*, Z*, NODESNOTCONNECTED (NR)
	RSODSOBJ	0*, Z*	*U
	RSODSPART	DATA, DEFINITION	

A.1.4 Authorizations Required for LKM SAP BW Upload (Background mode)

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

Note: Developer Key is required in this authorization.

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
S_RFC	ACTVT	16	
	RFC_NAME	RFC1, SDIFRUNTIME, SYST, ZODI_FGR, ZODI_FGR_PROD*	
	RFC_TYPE	FUGR	
S_TCODE	TCD	SE38, SU53	
S_BTCH_ADM	BTCADMIN	Y	
S_BTCH_JOB	JOBACTION	RELE	
	JOBGROUP	'	
S_CTS_ADMI	CTS_ADMFCT	TABL	
S_DATASET	ACTVT	06, 33, 34	

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
	FILENAME	*\\DEL-7TMK2BS\o diagentbox*	
	PROGRAM	SAPLSTRF, ZODI*	
S_DEVELOP	ACTVT	01	02,03,06,16
	DEVCLASS		
	OBJNAME	ZODI_FGR_PROD*	
	OBJTYPE	FUGR	PROD
	P_GROUP		
S_PROGRAM	P_ACTION	NR	EDIT, SUBMIT, VARIANT
	P_GROUP	NR	
S_TRANSPRT	ACTVT	01,02,03	03 (NR)
	TTYPE	DTRA, TASK	
S_RS_AUTH	BIAUTH	0*	0*
S_RS_ICUBE	ACTVT	03,06	NR
	RSICUBEOBJ	DATA, DEFINITION	NR
	RSINFOAREA	NODESNOTCONNECTE D	NR
	RSINFOCUBE	0*, Z*	NR

A.1.5 Authorizations Required for LKM SAP BW Execution (Background mode)

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

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
S_RFC	ACTVT	16	
	RFC_NAME	RFC1, SDIFRUNTIME, SLST, SYST, ZODI*	
	RFC_TYPE	FUGR	
S_TCODE	TCD	SU53	
S_BTCH_ADM	BTCADMIN	Y	
S_BTCH_JOB	JOBACTION	RELE	
	JOBGROUP	' '	
S_DATASET	ACTVT	06,33,34	
	FILENAME	\\10.30.0.201\OD I_SAPTEST*	
	PROGRAM	ZODI*	
S_DEVELOP	ACTVT	NR	01,02,03,06,07,1 6

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
	DEVCLASS	NR	
	OBJNAME	NR	S_TCODE
	OBJTYPE	NR	SUSO
	P_GROUP	NR	
S_RS_AUTH	BIAUTH	NR	0*
S_RS_ICUBE	ACTVT	03	
	RSICUBEOBJ	DATA, DEFINITION	
	RSINFOAREA	NODESNOTCONNECTE D	0*, Z*, NODESNOTCONNECTE D (NR)
	RSINFOCUBE	0*, Z*	

A.1.6 Authorizations Required for LKM SAP BW Upload (Dialog mode)

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

Note: Developer Key is required in this authorization.

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
S_RFC	ACTVT	16	
	RFC_NAME	RFC1, SDIFRUNTIME, SYST, ZODI_FGR, ZODI_FGR_PROD*	
	RFC_TYPE	FUGR	
S_TCODE	TCD	SE38, SU53	
S_CTS_ADMI	CTS_ADMFCT	TABL	
S_DATASET	ACTVT	06, 33, 34	
	FILENAME	*\\DEL-7TMK2BS\o diagentbox*,	
	PROGRAM	SAPLSLOG*, SAPLSTRF*, SAPLZODI*	
S_DEVELOP	ACTVT	01, 02, 03, 06	
	DEVCLASS		
	OBJNAME	ZODI_FGR_PROD*	
	OBJTYPE	FUGR, PROG	
	P_GROUP		
S_PROGRAM	P_ACTION	NR	EDIT, SUBMIT, VARIANT
	P_GROUP	NR	

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
S_TRANSPRT	ACTVT	01	
	TTYPE	DTRA, TASK	
S_RS_AUTH	BIAUTH	NR	0*
S_RS_ICUBE	ACTVT	03	NR
	RSICUBEOBJ	DATA, DEFINITION	NR
	RSINFOAREA	NODESNOTCONNECTE D	NR
	RSINFOCUBE	0*, Z*	NR

A.1.7 Authorizations Required for LKM SAP BW Execution (Dialog mode)

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

Object	Field Name	Value (BW 3.5)	Value (BI 7.0)
S_RFC	ACTVT	16	
	RFC_NAME	RFC1, SDIFRUNTIME, SYST, ZODI_FGR, ZODI_FGR_PROD*	
	RFC_TYPE	FUGR	
S_TCODE	TCD	SU53	
S_DATASET	ACTVT	06, 33, 34	
	FILENAME	*\\DEL-7TMK2BS\o diagentbox*,	
	PROGRAM	SAPLSLOG*, SAPLSTRF*, SAPLZODI*	
S_RS_AUTH	BIAUTH	NR	0*
S_RS_ICUBE	ACTVT	03	
	RSICUBEBOB	DATA, DEFINITION	DATA (NR)
	RSINFOAREA	NODESNOTCONNECTE D	NR
	RSINFOCUBE	0*, Z*	NR
S_RS_IOBJ	ACTVT	NR	03
	RSIOBJ	NR	
	RSIOBJCAT	NR	
	RSIOBJPART	NR	DEFINITION
S_RS_ODSO	ACTVT	NR	03
	RSINFOAREA	NR	
	RSODSOBJ	NR	
	RSODSPART	NR	DEFINITION

A.2 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 [Section B.2, "SAP Connection Test"](#) for more information.

A.2.1 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. These are installed during first execution of the RKM (UPLOAD_ABAP_CODE and UPLOAD_ABAP_BASE set to true).

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

A.3 Updating ODI SAP Components

During first-time installation the RKM installs some ODI objects into the SAP system. This installation consists of two parts: some base objects and some RFCs. When the RKM options `UPLOAD_ABAP_BASE` and `UPLOAD_ABAP_CODE` are both set to `true`, the base objects and the RFCs are installed. Such full installation requires that no ODI SAP objects are installed in the SAP system.

See [Uninstalling ODI SAP Components](#) for information about how to uninstall ODI SAP Components if needed.

If the ODI objects installed into your SAP systems have been installed by RKM SAP BW v23 or later, there is no need for reinstalling the base objects and it is sufficient to update just the RFCs. Reinstalling the RFCs is achieved by executing a reverse engineering with the RKM option `UPLOAD_ABAP_BASE` set to `false` and RKM option `UPLOAD_ABAP_CODE` set to `true`.

A.4 Uninstalling ODI SAP Components

For removing ODI connectivity from the SAP system, please follow the steps described in this section.

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

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

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

If it does not allow deleting the development class individually, delete all the objects one by one and then delete the development class.

8. Go to transaction SE10. Select **Modifiable** and **Workbench Requests**. Click **Display**.
9. Select the transport requests which have the ODI Objects and Release them.
10. Now repeat the steps 3 to 7 to delete Development Class.
11. Go to transaction SE01. Select **Modifiable** and **Workbench Requests**. Click **Display**. Release the Transport Request which has `ZODIBW_PCKG`.
12. Repeat the above steps for the `ZODI_LKM_PCKG` package. This step applies only to the installations done by RKM SAP BW v23 or later.

Additional Information for SAP ABAP ERP Adapter

This appendix describes the privileges that are required for connecting to SAP System, how you can test the connection outside of ODI using a standalone java utility, and how to uninstall SAP components.

This appendix includes the following sections:

- [Section B.1, "SAP ABAP ERP Required Privileges"](#)
- [Section B.2, "SAP Connection Test"](#)
- [Section B.3, "SAP Stand-Alone Connection Test"](#)
- [Section B.4, "Updating ODI SAP Components"](#)
- [Section B.5, "Uninstalling ODI SAP Components"](#)

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

B.1 SAP ABAP ERP Required Privileges

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

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

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

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

Table 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

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

B.1.1 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**.

B.1.2 Authorizations Required for RKM SAP ERP Upload

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

Note: Developer key is required in this authorization.

Object	Field Name	Value (4.6c)	Value (4.7)	Value (ECC 5)	Value (ECC 6)
S_RFC	ACTVT	16			
	RFC_NAME	RFC1, SDIFRUNTIME, SUTL, SYST, ZODI_FGR			
	RFC_TYPE	FUGR			
S_TCODE	TCD	SU53, SE38			
S_ADMI_FCD	S_ADMI_FCD	MEMO, SM02		SM02 (NR)	NR
S_CTS_ADMI	CTS_ADMFCT	TABL			
S_DATASET	ACTVT	34			
	FILENAME	\\HYPTTEST01\ sapmnt*			
	PROGRAM	SAPLSLOG, SAPLSTRF			
S_TABU_DIS	ACTVT	03			NR
	DICBERCLS	MA			NR
S_DEVELOP	ACTVT	01, 02, 03			16
	DEVCLAS	\$TMP, ZODI_ LKM_PCKG, ZODI_RKM_ PCKG			

Object	Field Name	Value (4.6c)	Value (4.7)	Value (ECC 5)	Value (ECC 6)
	OBJNAME	Z\$\$\$XRFC, ZODI_FGR, ZODI_FGR_ PROD, ZODI_ LKM_PCKG, ZODI_RKM_ PCKG			
	OBJTYPE	DEVC, FUGR, PR OG			
	P_GROUP				
S_TRANSPRT	ACTVT	01, 03	03 (NR)		
	TTYPE	DTRA, TASK			

B.1.3 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 (4.6c)	Value (4.7)	Value (ECC 5)	Value (ECC 6)
S_RFC	ACTVT	16			
	RFC_NAME	RFC1, SDIF, SY ST, ZODI_FGR	SDIFRUNTIME, SDIF (NR)	SDIFRUNTIME, SDIF (NR)	SDIFRUNTIME, SDIF (NR)
	RFC_TYPE	FUGR			
S_TCODE	TCD	SU53			
S_TABU_DIS	ACTVT	03			
	DICBERCLS	MA		CA, SC	

B.1.4 Authorizations Required for LKM SAP ERP Upload (Background mode)

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

Note: Developer Key is required in this authorization.

Object	Field Name	Value (4.6c)	Value (4.7)	Value (ECC 5)	Value (ECC 6)
S_RFC	ACTVT	16			
	RFC_NAME	RFC1, SDIF, SLST, SYST, ZODI*, ZODI_ FGR, ZODI_ FGR_PROD*	SDIFRUNTIME, SDIF (NR)	SDIFRUNTIME, SDIF (NR)	SDIFRUNTIME, SDIF (NR)
	RFC_TYPE	FUGR			
S_TCODE	TCD	SU53	SE38	SE38	NR
S_BTCH_ADM	BTCADMIN	Y			
S_BTCH_JOB	JOBACTION	RELE			

Object	Field Name	Value (4.6c)	Value (4.7)	Value (ECC 5)	Value (ECC 6)
	JOBGROUP	'			
S_CTS_ADMIN	CTS_ADMFCT	TABL			
S_DATASET	ACTVT	06,33,34			
	FILENAME	\\HYPTTEST01\ sapmnt*			
	PROGRAM	SAPLSLOG, SAPLSTRF, SAPLZODI*, ZODI*			
S_DEVELOP	ACTVT	01	02,03,06,16	02,03,06,16	
	DEVCLASS				
	OBJNAME	ZODI_FGR_ PROD*	ZANU*	ZANU*	
	OBJTYPE	FUGR		PROG	
	P_GROUP				
S_PROGRAM	P_ACTION	NR	EDIT, SUBMIT, VARIANT		NR
	P_GROUP	NR			NR
S_TRANSPRT	ACTVT	01,02,03		03 (NR)	03 (NR)
	TTYPE	DTRA,TASK			

B.1.5 Authorizations Required for LKM SAP ERP Execution (Background mode)

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

Object	Field Name	Value (4.6c)	Value (4.7)	Value (ECC 5)	Value (ECC 6)
S_RFC	ACTVT	16			
	RFC_NAME	RFC1, SDIFRUNTIME, SYST, ZODI_ FGR_PROD*	SLST	SLST	SLST
	RFC_TYPE	FUGR			
S_TCODE	TCD	SU53			
S_BTCH_ADM	BTCADMIN	Y			
S_BTCH_JOB	JOBACTION	RELE			
	JOBGROUP	' '			
S_DATASET	ACTVT	06,33,34			
	FILENAME	\\HYPTTEST01\ sapmnt*			
	PROGRAM	SAPLSLOG, SAPLSTRF, SAPLZODI*, ZODI*	(SAPLSLOG, SAPLSTRF, SAPLZODI*) NR	(SAPLSLOG, SAPLSTRF, SAPLZODI*) NR	(SAPLSLOG, SAPLSTRF, SAPLZODI*) NR

B.1.6 Authorizations Required for LKM SAP ERP Upload (Dialog mode)

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

Note: Developer Key is required in this authorization.

Object	Field Name	Value (4.6c)	Value (4.7)	Value (ECC 5)	Value (ECC 6)
S_RFC	ACTVT	16			
	RFC_NAME	RFC1, SDIFRUNTIME, SYST, ZODI_ FGR, ZODI_ FGR_PROD*			
	RFC_TYPE	FUGR			
S_TCODE	TCD	SU53, SE38			
S_DATASET	ACTVT	06, 33, 34			
	FILENAME	\\HYPTTEST01\ sapmnt*			
	PROGRAM	SAPLSLOG, SAPLSTRF, SAPLZODI*, ZODI*			
S_DEVELOP	ACTVT	01	02, 03, 06	02, 03, 06	
	DEVCLASS				
	OBJNAME	ZODI_FGR_ PROD*			
	OBJTYPE	FUGR	PROG		
S_PROGRAM	P_GROUP				
	P_ACTION	NR	EDIT, SUBMIT, VARIANT	EDIT, SUBMIT, VARIANT	NR
	P_GROUP	NR			
S_TRANSPRT	ACTVT	01			
	TTYPE	DTRA, TASK			

B.1.7 Authorizations Required for LKM SAP ERP Execution (Dialog mode)

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

Object	Field Name	Value (4.6c)	Value (4.7)	Value (ECC 5)	Value (ECC 6)
S_RFC	ACTVT	16			
	RFC_NAME	RFC1, SDIF, SY ST, ZODI*	SDIFRUNTIME, SDIF (NR)	SDIFRUNTIME, SDIF (NR)	SDIFRUNTIME, SDIF (NR)

Object	Field Name	Value (4.6c)	Value (4.7)	Value (ECC 5)	Value (ECC 6)
	RFC_TYPE	FUGR			
S_TCODE	TCD	SU53			
S_DATASET	ACTVT	06,33,34			
	FILENAME	*\DEL-7TMK2 BS\odiagentb ox*, \DEL-7TMK2B S\odiagentbo x			
	PROGRAM	SAPLSLOG*, SAPLSTRF*, SAPLZODI*			

B.2 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 *Developer's Guide for 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.

B.3 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. These are installed during first execution of the RKM (`UPLOAD_ABAP_BASE` and `UPLOAD_ABAP_CODE` set to `true`).

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

B.4 Updating ODI SAP Components

During first-time installation the RKM installs some ODI objects into the SAP system. This installation consists of two parts: some base objects and some RFCs. When the RKM options `UPLOAD_ABAP_BASE` and `UPLOAD_ABAP_CODE` are both set to `true`, the base objects and the RFCs are installed. Such full installation requires that no ODI SAP objects are installed in the SAP system.

See [Uninstalling ODI SAP Components](#) for information about how to uninstall ODI SAP Components if needed.

If the ODI objects installed into your SAP systems have been installed by RKM SAP ERP v32 or later, or RKM SAP BW v23 or later, there is no need for reinstalling the base objects and it is sufficient to update just the RFCs. Reinstalling the RFCs is achieved by executing a reverse engineering with the RKM option `UPLOAD_ABAP_BASE` set to `false` and RKM option `UPLOAD_ABAP_CODE` set to `true`.

B.5 Uninstalling ODI SAP Components

For removing ODI connectivity from the SAP system, please follow the steps described in this section.

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

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

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

If it does not allow deleting the development class individually, delete all the objects one by one and then delete the development class.

8. Go to transaction SE10. Select **Modifiable** and **Workbench Requests**. Click **Display**.
9. Select the transport requests which have the ODI Objects and Release them.
10. Now repeat the steps 3 to 7 to delete Development Class.
11. Go to transaction SE01. Select **Modifiable** and **Workbench Requests**. Click **Display**. Release the Transport Request which has ZODI_DEVCLASS.
12. Repeat the above steps for package name ZODI_LKM_PCKG. This applies only to the installations done by RKM SAP ERP v32 or later.