

Oracle® Fusion Middleware

Application Adapters Guide for Oracle Data Integrator

11g Release 1 (11.1.1)

E17466-07

February 2013

Copyright © 2011, 2013, 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 RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

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

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

This software or hardware and documentation may provide access to or information 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	ix
Audience	ix
Documentation Accessibility	ix
Related Documents	ix
Conventions	x
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 an Interface	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 EnterpriseOne	
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.....	3-4
3.3	Setting Up the Topology	3-4
3.3.1	Create a Data Server	3-5
3.3.2	Create a Physical Schema	3-5
3.4	Setting up an Integration Project	3-5
3.5	Creating and Reverse-Engineering a Model.....	3-6
3.5.1	Create a Model	3-6
3.5.2	Reverse-Engineer JDE Tables.....	3-6
3.6	Designing an Interface.....	3-7
3.6.1	Loading Data from JDE.....	3-7
3.6.2	Integrating Data in JDE.....	3-8

4 JD Edwards World

4.1	Introduction	4-1
4.1.1	Concepts.....	4-1
4.1.2	Knowledge Modules	4-1
4.2	Installation and Configuration.....	4-2
4.2.1	System Requirements and Certifications	4-2
4.2.2	Technology Specific Requirements	4-2
4.2.3	Connectivity Requirements.....	4-3
4.3	Setting Up the Topology	4-3
4.3.1	Create a Data Server	4-3
4.3.2	Create a Physical Schema	4-4
4.4	Setting up an Integration Project	4-4
4.5	Creating and Reverse-Engineering a Model.....	4-4
4.5.1	Create a Model	4-4
4.5.2	Reverse-Engineer JDE Tables.....	4-5
4.6	Designing an Interface.....	4-5
4.6.1	Loading Data from JDE.....	4-6
4.6.2	Integrating Data in JDE.....	4-6

5 Oracle PeopleSoft

5.1	Introduction	5-1
5.1.1	Concepts.....	5-1
5.1.2	Knowledge Modules	5-1
5.2	Installation and Configuration.....	5-2
5.2.1	System Requirements and Certifications	5-2
5.2.2	Technology Specific Requirements	5-2
5.2.3	Connectivity Requirements.....	5-2
5.3	Setting up the Topology	5-2
5.3.1	Create a Data Server	5-2
5.3.2	Create a Physical Schema	5-3
5.4	Setting up the Project.....	5-3
5.5	Creating and Reverse-Engineering a Model.....	5-3

5.5.1	Create a Model	5-3
5.5.2	Reverse-Engineer PeopleSoft Tables.....	5-4
5.6	Designing an Interface	5-4
5.6.1	Loading Data from PeopleSoft.....	5-4

6 Oracle Siebel CRM

6.1	Introduction	6-1
6.1.1	Concepts.....	6-1
6.1.2	Overview of Extracting Data from Siebel	6-1
6.1.3	Knowledge Modules	6-2
6.2	Installation and Configuration.....	6-3
6.2.1	System Requirements and Certifications	6-3
6.2.2	Technology Specific Requirements	6-3
6.2.3	Connectivity Requirements.....	6-4
6.3	Setting up the Topology	6-4
6.3.1	Create a Data Server	6-4
6.3.2	Create a Physical Schema	6-4
6.4	Setting up the Project.....	6-4
6.5	Creating and Reverse-Engineering a Model.....	6-5
6.5.1	Create a Model	6-5
6.5.2	Reverse-Engineer Siebel CRM Tables.....	6-5
6.6	Designing an Interface	6-6
6.6.1	Loading Data from Siebel.....	6-6
6.6.2	Loading Data from Siebel through EIM tables.....	6-6
6.6.3	Integrating Data in Siebel through EIM tables.....	6-7

7 SAP ABAP BW

7.1	Introduction	7-1
7.1.1	Concepts.....	7-1
7.1.2	Knowledge Modules	7-2
7.1.3	Overview of the SAP BW Integration Process.....	7-2
7.1.3.1	Reverse-Engineering Process.....	7-3
7.1.3.2	Integration Process	7-3
7.2	Installation and Configuration.....	7-3
7.2.1	System Requirements and Certifications	7-4
7.2.2	Technology Specific Requirements	7-4
7.2.3	Connectivity Requirements.....	7-5
7.2.3.1	Installing and Configuring JCo	7-5
7.2.3.2	Gathering SAP Connection Information.....	7-6
7.2.3.3	Gathering FTP Connection Information	7-7
7.2.3.4	Gathering Shared Directory Information.....	7-7
7.2.4	Getting the Right Privileges	7-7
7.3	Defining the Topology	7-7
7.3.1	Create the File Data Server	7-7
7.3.1.1	Create a File Data Server	7-7
7.3.1.2	Create the File Schema.....	7-8

7.3.2	Create the SAP ABAP Data Server.....	7-9
7.3.2.1	Create the SAP ABAP Data Server.....	7-9
7.3.2.2	Create the SAP ABAP Schema.....	7-10
7.4	Setting up the Project.....	7-10
7.5	Creating and Reverse-Engineering a Model.....	7-11
7.5.1	Creating a SAP BW Model	7-11
7.5.2	Reverse-Engineering a SAP BW Model.....	7-11
7.6	Designing an Interface.....	7-12
7.7	Considerations for SAP BW Integration.....	7-12
7.7.1	File Transfer Configurations	7-12
7.7.1.1	Transfer using a Shared Directory (recommended).....	7-12
7.7.1.2	FTP based Transfer.....	7-15
7.7.2	Controlling ABAP Uploading / ABAP code in production	7-17
7.7.3	Managing ODI SAP Transport Requests.....	7-18
7.7.4	Log Files	7-18
7.7.5	Limitation of the SAP BW Adapter.....	7-19

8 SAP ABAP ERP

8.1	Introduction	8-1
8.1.1	Concepts.....	8-1
8.1.2	Knowledge Modules	8-2
8.1.3	Overview of the SAP ABAP Integration Process	8-2
8.1.3.1	Reverse-Engineering Process.....	8-2
8.1.3.2	Integration Process	8-3
8.2	Installation and Configuration.....	8-3
8.2.1	System Requirements and Certification.....	8-3
8.2.2	Technology Specific Requirements	8-3
8.2.3	Connectivity Requirements.....	8-4
8.2.3.1	Installing and Configuring JCo	8-5
8.2.3.2	Gathering SAP Connection Information.....	8-5
8.2.3.3	Gathering FTP Connection Information	8-6
8.2.3.4	Gathering Shared Directory Information.....	8-6
8.2.4	Getting the Right Privileges	8-7
8.3	Defining the Topology	8-7
8.3.1	Create the File Data Server	8-7
8.3.1.1	Create a File Data Server	8-7
8.3.1.2	Create the File Schema.....	8-8
8.3.2	Create the SAP ABAP Data Server.....	8-9
8.3.2.1	Create the SAP ABAP Data Server.....	8-9
8.3.2.2	Create the SAP ABAP Schema.....	8-10
8.4	Setting up the Project.....	8-10
8.5	Creating and Reverse-Engineering a Model.....	8-10
8.5.1	Creating a SAP ERP Model	8-11
8.5.2	Reverse-Engineering a SAP ERP Model.....	8-11
8.6	Designing an Interface.....	8-11
8.7	Considerations for SAP ERP Integration.....	8-12
8.7.1	File Transfer Configurations	8-12

8.7.1.1	Transfer using a Shared Directory (recommended).....	8-12
8.7.1.2	FTP based Transfer.....	8-14
8.7.2	Controlling ABAP Uploading	8-16
8.7.3	Managing ODI SAP Transport Requests.....	8-17
8.7.4	Log Files	8-18
8.7.5	Limitations of the SAP ABAP Adapter	8-18

A Additional Information for SAP ABAP BW Adapter

A.1	SAP ABAP BW Required Privileges	A-1
A.1.1	Authorizations Required for RKM SAP BW Setup	A-1
A.1.2	Authorizations Required for RKM Execution	A-2
A.1.3	Authorizations Required for LKM Execution	A-2
A.1.4	Authorizations Required for LKM Execution for Production	A-2
A.2	SAP Stand-Alone Connection Test.....	A-2
A.3	SAP Stand-Alone Connection Test.....	A-2
A.4	Uninstalling ODI SAP Components.....	A-3

B Additional Information for SAP ABAP ERP Adapter

B.1	SAP ABAP ERP Required Privileges	B-1
B.1.1	Authorizations Required for RKM SAP ERP Setup	B-1
B.1.2	Authorizations Required for RKM Execution	B-4
B.1.3	Authorizations Required for LKM Execution	B-5
B.1.4	Authorizations Required for LKM Execution for Production	B-7
B.2	SAP Connection Test	B-9
B.3	SAP Stand-Alone Connection Test.....	B-9
B.4	Uninstalling ODI SAP Components.....	B-10

C Installing the librfc32/librfc64 Library

C.1	Installing the librfc32/librfc64 Library	C-1
-----	--	-----

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:

- *Oracle Fusion Middleware Getting Started with Oracle Data Integrator*
- *Oracle Fusion Middleware Installation Guide for Oracle Data Integrator*
- *Oracle Fusion Middleware Upgrade Guide for Oracle Data Integrator*
- *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*
- *Oracle Fusion Middleware Connectivity and Modules Guide for Oracle Data Integrator*

- *Oracle Fusion Middleware Knowledge Module Developer's Guide for Oracle Data Integrator*
- *Oracle Data Integrator 11g Online Help*
- *Oracle Data Integrator 11g Release Notes, included with your Oracle Data Integrator 11g 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

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](#)
- [JD Edwards EnterpriseOne](#)
- [JD Edwards World](#)
- [Oracle PeopleSoft](#)
- [Oracle Siebel CRM](#)
- [SAP ABAP BW](#)
- [SAP ABAP ERP](#)

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" and "JMS" in the *Oracle Fusion Middleware Connectivity and 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 an Interface"](#)

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 interface in a batch. Use this Group ID in subsequent interfaces. 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 *Oracle Fusion Middleware 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 *Oracle Fusion Middleware 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 *Oracle Fusion Middleware 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 an interface.

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 *Oracle Fusion Middleware 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 *Oracle Fusion Middleware 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 *Oracle Fusion Middleware 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 and columns, as well as the primary and foreign keys in the datastores
- 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 *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*. This section details only the fields specific to EBS tables:

1. In the Reverse 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 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 integration interfaces.

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 and columns, as well as the primary and foreign keys are reversed in the datastores.
- 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 an Interface

You can use E-Business Suite as a source and a target of an integration interface.

The KM choice for an interface determines the abilities and performance of this interface. 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 integration interfaces 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 *Oracle Fusion Middleware 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 *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for more information about the IKM Oracle Incremental Update.

The configuration of integration interfaces 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 integration interface that loads a group of interface tables in one single batch.

To create a Group ID in an integration interface:

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 integration interface 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 integration interfaces belonging to a batch, you must use an existing Group ID.

Using an existing Group ID

To use an existing Group ID in an integration interface:

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 integration interface 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 integration interface 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 integration interface 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 integration interfaces, 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 integration interface, if the Open Interface only contains a single table. The execution of the Open Interface program is started in the last integration interface of a package. This integration interface 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 EnterpriseOne

This chapter describes how to work with JD Edwards EnterpriseOne 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 an Interface"](#)

3.1 Introduction

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

3.1.1 Concepts

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

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

3.1.2 Knowledge Modules

Oracle Data Integrator provides the Knowledge Modules listed in [Table 3-1](#) for handling JDE EnterpriseOne data. These specific JDE KMs provide connectivity and integration of the JDE EnterpriseOne platform with any database application through Oracle Data Integrator.

Table 3–1 JDE Knowledge Modules

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

3.2 Installation and Configuration

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

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

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

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

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

1. Set up and configure the federated server and database. Configuring the federated server to access DB2 data sources involves supplying the server with information

about the DB2 data sources and objects you want to access. You can configure access to DB2 data sources two ways:

- Through the DB2 Control Center
 - Through the DB2 Command Center or command line processor (CLP)
2. Add a DB2 data source to a federated server:
 1. Catalog a node entry in the federated node directory.
For example, if TCP/IP is your communication protocol issue the following command:

```
CATALOG TCPIP NODE <db2node> REMOTE <system> SERVER
<server_name>
```
 2. Catalog the remote database in the federated system database director using the following command:

```
CATALOG DATABASE <db_name> AS <alias_name> AT NODE <db2_
node> AUTHENTICATION SERVER
```
 3. Create the wrapper using the following command:

```
CREATE WRAPPER DRDA
```

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

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

where:

```
AUTHORIZATION <user>
```

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

```
PASSWORD <password>
```

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

```
DBNAME <db_name>
```

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

Although the database name is specified as an option in the CREATE SERVER statement, it is required for DB2 data sources.
 5. Create the user mappings. If a user's authorization ID to access the federated database differs from the user's authorization ID to access a data source, you need to define a user mapping between the two authorization IDs.

```
CREATE USER MAPPING FOR <db2user> SEVER <server_name>
OPTIONS (REMOTE_AUTHID <remote_user> REMOTE_PASSWORD
<remote_password>)
```

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

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

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

Create the nicknames as described in [Table 3-2](#).

Table 3-2 Nicknames for Table Objects

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

Note: Even if the table names and schemas might be different between JDE version 8 and 9 and differ from the values listed in [Table 3-2](#), the nicknames should stay the same.

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

3.2.3 Connectivity Requirements

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

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

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 data.

3.3.1 Create a Data Server

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

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

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

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

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 *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

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

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

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

3.4 Setting up an Integration Project

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

Import the following KMs into your Oracle Data Integrator project:

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

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

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

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 technology hosting the JDE tables and on the logical schema created when configuring the JDE Connection using the standard procedure, as described in "Creating a Model" of the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

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

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

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

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

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

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

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

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

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

3.6 Designing an Interface

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

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

3.6.1 Loading Data from JDE

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

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

- "Oracle Database"
- "Generic SQL"

- "Microsoft SQL Server"
- "IBM DB2 for iSeries"
- "IBM DB2 UDB"

3.6.2 Integrating Data in JDE

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

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

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

These interfaces should be sequenced in a package.

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

To create an interface targeting JDE:

1. Create an integration interface with Z-tables as target datastores.
2. Create joins, filters, and mappings as usual.
3. In the Flow tab select the **IKM JDE Enterprise One Control Append (UBE)**.
4. Set the standard KM options (INSERT, COMMIT, FLOW_CONTROL).
5. If this interface launches the RunUBE command, specify the KM options as follows:
 1. Set the JDE_RUNUBE option to Yes.
 2. Specify the JDE_DIRECTORY in which the RunUBE command is executed.
 3. If you want to create a password file, set the password related options as shown in [Table 3-3](#).

Table 3-3 Password Related KM Options

Option	Value	Notes
JDE_CREATE_PWD_FILE	Yes	To enhance RunUBE security in a Unix or iSeries environment, when the RunUBE command is submitted, the system reads the text file specified in the JDE_PWD_FILE option and uses the JD Edwards EnterpriseOne user ID and password as indicated in the text file.

Table 3–3 (Cont.) Password Related KM Options

Option	Value	Notes
JDE_PWD_FILE	Absolute path of the password security file	This file contains the user id and password specified in the JDE_USER_ID and JDE_PWD options.
JDE_DELETE_PWD_FILE	E D	Enter D to delete the password file. Enter F to keep the password file. Note that even if the password file is removed after the execution of the command, the file should be kept in a secure location on the server file system
JDE_USER_ID	JDE EnterpriseOne user ID	The user must have permissions to run the report.
JDE_PWD	JDE EnterpriseOne password	The EnterpriseOne password that corresponds to the user ID.

4. Set the parameters for the RunUBE command as shown in [Table 3–4](#).

Table 3–4 RunUBE Command related KM Options

Option	Value	Notes
JDE_ENVIRONMENT	The JDE EnterpriseOne environment	
JDE_ROLE	The JDE EnterpriseOne role	
JDE_REPORT	The system name of the report that you want to process	For example: APS Outbound Processor (R34A400) and APS Inbound Processor (R34A410) for flat files, and the APS SCBM 2.0 Outbound Processor (R34A700) and APS Master Inbound Processor (R34A820) for XML files
JDE_VERSION	The name of the version of the report that you want to process	For example: XJDE0001. Note that you must enter a version name; you cannot submit the template of a report.
JDE_JOB_QUEUE	The name of the job queue to which the system should route the batch job	For example: QBATCH
JDE_PROCESSING_MODE	The processing mode	Enter B to use batch processing. In this case, the system uses the Job Control Status Master table (F986110) to assign the report a place in the queue. Enter I to use the interactive mode. This mode runs the report immediately outside of the JDE EnterpriseOne queuing mechanism

Table 3–4 (Cont.) RunUBE Command related KM Options

Option	Value	Notes
JDE_HOLD_CODE	The hold code	Enter P to send the output to a printer immediately after the job completes. Enter H to hold the processed file without printing. You can print the job later using the Work With Servers program (P986116) that is accessible from the System Administration Tools menu (GH9011).
JDE_SAVE_CODE	The save code	Enter S to save the file after processing is complete. The delete option (D) is reserved for future use. Currently, the delete option is disabled.

Limitations of the IKM JDE Enterprise One Control Append (UBE)

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

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 4.1, "Introduction"](#)
- [Section 4.2, "Installation and Configuration"](#)
- [Section 4.3, "Setting Up the Topology"](#)
- [Section 4.4, "Setting up an Integration Project"](#)
- [Section 4.5, "Creating and Reverse-Engineering a Model"](#)
- [Section 4.6, "Designing an Interface"](#)

4.1 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.

4.1.1 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)

4.1.2 Knowledge Modules

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

Table 4–1 JDE Knowledge Modules

Knowledge Module	Description
RKM JDE World	Reverse-engineers the metadata of the applications' objects such as tables and interface tables from JDE World installed on DB2 iSeries database.
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

4.2 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](#)

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

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

IKM JDE World Control Append

The RPG program requires the following files:

- `JDEWorldJDBC.jar` - This JDE World JDBC driver file must be in the `userlib` or `driver` folder.

See "Add Additional Drivers and Open Tools" in the *Oracle Fusion Middleware Installation Guide for Oracle Data Integrator* for more information about these folders.

- `WorldBase.jar` - This jar file must be in the `userlib` or `driver` folder.
- `config.xml` - This configuration file must be in the `ODI_HOME/odi/bin` folder.

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

Tip: The `ODI_HOME/misc/jde-world` directory also contains a `jdeworld_example.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 4.6.2, "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)
- Move it to a safe directory accessible by Oracle Data Integrator Studio or the standalone agent.

RKM JDE World

There are no technology specific requirements for the RKM JDE World.

4.2.3 Connectivity Requirements

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

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 *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator*:

4.3 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.

4.3.1 Create a Data Server

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

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 *Oracle Fusion Middleware 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.

4.3.2 Create a Physical Schema

Create a physical schema under the data server that you have created in [Section 4.3.1, "Create a Data Server"](#). Use the standard procedure, as described in "Creating a Physical Schema" of the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

This schema must point to the 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 an interface.

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

4.4 Setting up an Integration Project

Setting up a project using JDE World features follows the standard procedure. See "Creating an Integration Project" of the *Oracle Fusion Middleware 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

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 *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator*.

4.5 Creating and Reverse-Engineering a Model

This section contains the following topics:

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

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

4.5.2 Reverse-Engineer JDE Tables

The JDE RKM's are able to reverse-engineer JDE tables. These RKM's 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, use the usual procedure, as described in "Reverse-engineering a Model" of the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*. This section details only the fields specific to JDE World tables:

1. In the Reverse tab of the Model, select the RKM JDE World.
2. Set the RKM 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 integration interfaces.

4.6 Designing an Interface

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

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

4.6.1 Loading Data from JDE

After performing a reverse-engineering using the RKM JDE World, you can use JDE World data tables as a source of an integration interface 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 an integration interface. The generic SQL and IBM DB2 for iSeries KMs can be used for this purpose. See the following chapters in the *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for more information:

- "Generic SQL"
- "IBM DB2 for iSeries"

4.6.2 Integrating Data in JDE

After performing a reverse-engineering using the RKM JDE World, you can use JDE Z-tables as a target of an interface 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 interfaces, without calling the RPG program. These interfaces 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 interface loading the last required Z-table. This interface also uses the IKM JDE World Control Append with the JDE_INVOKE option set to `Yes`.

These interfaces 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 an interface targeting JDE World:

1. Create an integration interface with Z-tables as target datastores.
2. Create joins, filters, and mappings as usual.
3. In the Flow tab select the **IKM JDE World Control Append**.
4. Set the standard KM options (INSERT, COMMIT, FLOW_CONTROL).
5. If this interface 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 4-2](#).

Table 4–2 Connection Related KM Options

Option	Value	Notes
USE_SECURITY_FILE	Yes	To enhance security when the RPG program is submitted, the system reads the text file specified in the JDE_SECURITY_FILE option and uses the JD Edwards World user, password, environment, and address as indicated in the text file.
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 4.2.2, "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 4–3](#).

Table 4–3 RPG Program related KM Options

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

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

Option	Value	Notes
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.
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 interface.
- 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.
- 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 (WRKSPLF) command.

This chapter describes how to work with Oracle PeopleSoft 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, "Setting up the Topology"](#)
- [Section 5.4, "Setting up the Project"](#)
- [Section 5.5, "Creating and Reverse-Engineering a Model"](#)
- [Section 5.6, "Designing an Interface"](#)

5.1 Introduction

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

5.1.1 Concepts

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

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

5.1.2 Knowledge Modules

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

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

Table 5–1 PeopleSoft Knowledge Modules

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

5.2 Installation and Configuration

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

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

5.2.1 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>.

5.2.2 Technology Specific Requirements

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

5.2.3 Connectivity Requirements

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

5.3 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 PeopleSoft data.

5.3.1 Create a Data Server

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

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

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

5.3.2 Create a Physical Schema

Create a physical schema under the data server that you have created in [Section 5.3.1, "Create a Data Server"](#). Use the standard procedure, as described in "Creating a Physical Schema" of the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

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

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

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

5.4 Setting up the Project

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

Import the following KMs into your Oracle Data Integrator project:

- RKM PeopleSoft ORACLE
- RKM PeopleSoft MSSQL

In addition to these specific PeopleSoft KMs, import the standard LKMs for the Oracle or Microsoft SQL Server technologies. See Oracle Database "Knowledge Modules" and Microsoft SQL Server "Knowledge Modules" in the *Oracle Fusion Middleware 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:

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

5.5.1 Create a Model

Create a Model based on the Oracle or Microsoft SQL technology and on the logical schema created when setting up the topology using the standard procedure, as

described in "Creating a Model" of the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

5.5.2 Reverse-Engineer PeopleSoft Tables

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

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

The reverse-engineering process returns the following information:

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

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

1. In the Reverse tab of the Model, select the RKM PeopleSoft ORACLE or RKM PeopleSoft MSSQL.

2. Set the BUSINESS OBJECT RKM option as follows:

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

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

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

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

5.6 Designing an Interface

You can use PeopleSoft data tables as a source of an integration interface.

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

5.6.1 Loading Data from PeopleSoft

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

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

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

- "Oracle Database"
- "Generic SQL"
- "Microsoft SQL Server"

This chapter describes how to work with Oracle Siebel CRM Knowledge Modules in Oracle Data Integrator.

This chapter includes the following sections:

- [Section 6.1, "Introduction"](#)
- [Section 6.2, "Installation and Configuration"](#)
- [Section 6.3, "Setting up the Topology"](#)
- [Section 6.4, "Setting up the Project"](#)
- [Section 6.5, "Creating and Reverse-Engineering a Model"](#)
- [Section 6.6, "Designing an Interface"](#)

6.1 Introduction

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

6.1.1 Concepts

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

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

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

6.1.2 Overview of Extracting Data from Siebel

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

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

Data-level integration

Data extraction is performed directly on the Siebel Business Components tables. You can use a Siebel data model as a source of an integration interface by extracting data from the Siebel Database and integrate them into another system. Using Siebel as a source in these conditions is the same as using a regular table as a source in an integration interface. This integration method is read-only.

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

Integration through EIM tables

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

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

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

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

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

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

6.1.3 Knowledge Modules

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

Table 6–1 Siebel CRM Knowledge Modules

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

6.2 Installation and Configuration

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

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

6.2.1 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>.

6.2.2 Technology Specific Requirements

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

6.2.3 Connectivity Requirements

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

6.3 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 Siebel CRM data.

6.3.1 Create a Data Server

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

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

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

6.3.2 Create a Physical Schema

Create a physical schema under the data server that you have created in [Section 6.3.1, "Create a Data Server"](#). Use the standard procedure, as described in "Creating a Physical Schema" of the *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

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

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

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

6.4 Setting up the Project

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

Import the following KMs into your Oracle Data Integrator project:

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

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

6.5 Creating and Reverse-Engineering a Model

This section contains the following topics:

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

6.5.1 Create a Model

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

6.5.2 Reverse-Engineer Siebel CRM Tables

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

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

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

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

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

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

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

Table 6–2 KM options for Siebel EIM Knowledge Modules

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

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

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

6.6 Designing an Interface

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

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

6.6.1 Loading Data from Siebel

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

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

- "Oracle Database"
- "Generic SQL"
- "Microsoft SQL Server"

6.6.2 Loading Data from Siebel through EIM tables

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

- IFB_PATH: Specify the path where you want to create the EIM configuration file (.ifb).
- SRVRMGR_PATH: Specify the location of the Siebel srvmgr binary. This parameter is mandatory.
- SIEBEL_GATEWAY: Specify the network address of the Gateway Server machine.

- SIEBEL_ENTERPRISE: Indicate the name of the Enterprise Server.
- SIEBEL_SERVER: Indicate the name of the Siebel Server.
- SERVER_USER: Indicate the user name of the Server administrator.
- SERVER_USER_PWD: Indicate the Server administrator password.

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

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

6.6.3 Integrating Data in Siebel through EIM tables

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

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

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

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

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

This chapter includes the following sections:

- [Section 7.1, "Introduction"](#)
- [Section 7.2, "Installation and Configuration"](#)
- [Section 7.3, "Defining the Topology"](#)
- [Section 7.4, "Setting up the Project"](#)
- [Section 7.5, "Creating and Reverse-Engineering a Model"](#)
- [Section 7.6, "Designing an Interface"](#)
- [Section 7.7, "Considerations for SAP BW Integration"](#)

7.1 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 the *Oracle Data Integrator Getting Started with SAP ABAP BW Adapter* on Oracle Technical Network (OTN):

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

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

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

7.1.2 Knowledge Modules

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

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

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

Table 7–1 SAP BW Knowledge Modules

Knowledge Module	Description
LKM SAP BW to Oracle (SQLLDR)	Extracts data from SAP BW system into a flat file and then loads it into Oracle Staging Area using the SQL*LOADER command line utility.
RKM SAP ERP Connection Test	This RKM is used for testing the SAP connection from Oracle Data Integrator. See 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 ERP to SQL	Extracts data from SAP BW into a flat file and then loads it into a Staging Area using a JDBC connection.

7.1.3 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 ERP 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.

7.1.3.1 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.

7.1.3.2 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 interfaces 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 interface. 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 7.7.1, "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.

7.2 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](#)

7.2.1 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>.

7.2.2 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 7.2.1, "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 7.7.1, "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 interface (when this agent runs on the FTP server machine), or remotely (when this agent is located on a different machine than the FTP server). This FTP server must be accessible over the network from both the SAP BW machine and the agent machine.

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

7.2.3 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](#)

7.2.3.1 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_misc` directory (Studio) and into `<ODI_HOME>/oracledi/agent/drivers` (Standalone Agent).
 5. Restart the ODI Components using SAP (ODI Studio, Standalone Agent)

6. Check the JCo installation.

Note: Note: The SAP JCo software requires the `librfc32/librfc64` library, but does not include it. It is included e.g. with the SAP GUI software. In case this library is not yet installed, it has to be downloaded and installed following the steps outlined in [Appendix C, "Installing the librfc32/librfc64 Library."](#)

7.2.3.2 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.

7.2.3.3 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.

7.2.3.4 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.

7.2.4 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.

7.3 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](#)

7.3.1 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.

7.3.1.1 Create a File Data Server

Create a File data server as described in "Creating a File Data Server" of the *Oracle Fusion Middleware 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 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 *Oracle Fusion Middleware 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 7.7.1, "File Transfer Configurations"](#) for more information.

7.3.1.2 Create the File Schema

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

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

- For the FTP file transfer:
 - *Directory (Schema)*: Path on the FTP server to upload or download extraction files from the remote location. This path is used when uploading extraction files from the SAP BW system into the FTP server. It is also used by a remote agent to download the extraction files. Note that this path must use slashes and must end with a slash character.
 - *Directory (Work Schema)*: Local path on the FTP server's machine. This path is used by an agent installed on this machine to access the extraction files without passing via FTP. This access method is used if the FTP_TRANSFER_METHOD parameter of the LKM SAP BW to Oracle (SQLLDR)/ LKM SAP BW to SQL is set to NONE. The Work Schema is a local directory location. As a consequence, slashes or backslashes should be used according to the operating system. This path must end with a slash or backslash.

Path names given on Data and Work schemas are not necessarily the same: the FTP server may provide access to a FTP directory named `/sapfiles` - the

value for Directory (Schema) - while the files are accessed locally in
 c:\inetpub\ftproot\sapfiles - the value for Directory (Work Schema).

- For the Shared Directory transfer:
 - *Directory (Schema)*: Path (UNC) of the shared folder to write and read extraction files. SAP System writes the extraction files into this folder. It is also used by a remote agent to copy the extraction files to the ODI agent machine. Note that this path must use slashes or backslashes according to the operating system of the SAP Application Server and must end with a slash or backslash character.
 - *Directory (Work Schema)*: Local path on the server's machine hosting the shared folder. This path is used by an agent installed on this machine to access the extraction files without passing through the shared folder. This access method is used if the FTP_TRANSFER_METHOD parameter of the LKM SAP BW to Oracle (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 7.7.1, "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`.

7.3.2 Create the SAP ABAP Data Server

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

7.3.2.1 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 *Oracle Fusion Middleware 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 7.7.2, "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 7.7.2, "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.

7.3.2.2 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 *Oracle Fusion Middleware 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 *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator* in the appropriate context.

7.4 Setting up the Project

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

Import the following KMs into your Oracle Data Integrator project:

- RKM 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 *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for a list of available KMs.

7.5 Creating and Reverse-Engineering a Model

This section contains the following topics:

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

7.5.1 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 *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

7.5.2 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 *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*. This section details only the fields specific to SAP BW:

1. In the Reverse 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.

4. *For first time RKM use only:* Set UPLOAD_ABAP_CODE 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 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.

7.6 Designing an Interface

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

1. Create an interface with source datastores from the SAP BW Model. This interface should have an Oracle target or use an Oracle schema as the Staging Area.
2. Create joins, filters and mappings for your interface.
3. In the Flow tab of the interface, select the source set containing the SAP BW source data object(s) and select the LKM SAP BW to Oracle (SQLLDR).

7.7 Considerations for SAP BW Integration

This section includes the following topics:

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

7.7.1 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](#)

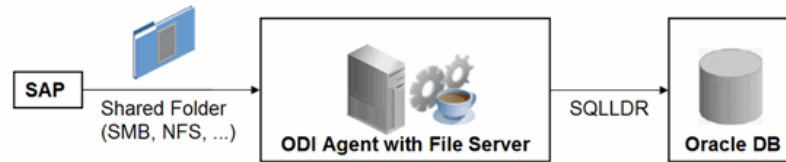
7.7.1.1 Transfer using a Shared Directory (recommended)

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 7-1 Configuration 1



This image shows the configuration where the shared directory is physically located on the ODI Agent machine.

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 (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.

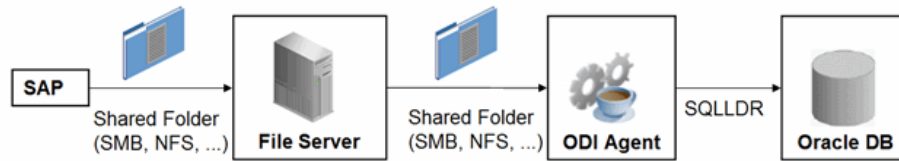
This configuration requires the following Topology settings:

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

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

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

Figure 7-2 Configuration 2



This image shows the configuration where the shared directory is not physically located on the ODI Agent machine.

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`.

7.7.1.2 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 7-3 Configuration 1: FTP Server is installed on an ODI Agent machine



This image shows the file transfer configuration where the FTP server is installed on an ODI Agent machine.

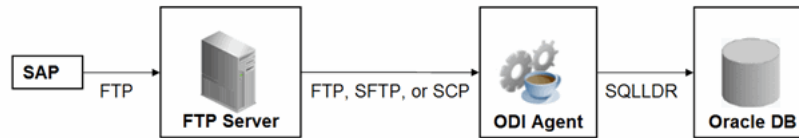
The configuration shown in Figure 7-3 is used, when `FTP_TRANSFER_METHOD = NONE`. In this configuration the following data movements are performed:

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

This configuration requires the following Topology settings:

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

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



This image shows the file transfer configuration where the FTP server is not installed on an ODI Agent machine.

The configuration shown in Figure 7–4 is used, when `FTP_TRANSFER_METHOD` is `FTP`, `SFTP` or `SCP`. In this configuration the following data movements are performed:

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

This configuration requires the following Topology settings:

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

Considerations and Limitations:

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

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

7.7.2 Controlling ABAP Uploading / ABAP code in production

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

Once an Interface 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 *Oracle Fusion Middleware 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 an interface 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 interfaces
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 interface/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 Interface's ABAP extraction code and not to extract any data, such as for an automated installation. In this case all interfaces can be linked inside a package with the KM option `EXECUTE_ABAP_CODE` set to `False` in every interface. Executing this package will then install all ABAP code, but will not perform any execution.

To avoid the modification of all interfaces (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.

7.7.3 Managing ODI SAP Transport Requests

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

An ODI interface 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 interface are assigned to one SAP function group. The ABAP extraction programs for a different interface will be assigned to a different SAP function group. The default function group name is similar to ZODI_FGR_<Interface Id>.

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

When the ABAP code of multiple ODI Interfaces should be grouped into a single transport request for more coarse-grained deployment control, the KM option SAP_FUNCTION_GROUP_NAME for all LKMs in these interfaces 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_<Interface Id>_<SourceSet Id>. This ensures convenient development due to unique program names. While the InterfaceId never changes, certain changes to an ODI interface 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.

7.7.4 Log Files

During the RKM and LKM execution several log files are created. These log files may contain valuable details for troubleshooting. [Table 7-2](#) describes the different log files and their usage:

Table 7-2 Log Files

Default Log File Name	KM / Phase	Content
<System Temp Dir>/sap_rkm_bw_<ODI Session Number>.log	RKM	Execution Log of metadata retrieval
<System Temp Dir>/sap_rkm_bw_<ODI Session Number>.log.opentool.log	RKM	Information about first time installation of SAP RFC for RKM
<System Temp Dir>/ODI_BW_Log/ODI_<Interface Id>_<SrcSet>.genlog	LKM - Generation Time	Information about code generation for ABAP extractor

Table 7–2 (Cont.) Log Files

Default Log File Name	KM / Phase	Content
<System Temp Dir>/ODI_BW_ Log/SAPAbapExecuteOpenTool_ <Interface Id>_<SrcSet>.log	LKM - Runtime	Information about installation of ABAP extractor
<System Temp Dir>/ODI_BW_ Log/SAPAbapExecuteOpenTool_ <Interface Id>_<SrcSet>.log	LKM - Runtime	Information about Delta Extraction
<System Temp Dir or local FTP dir>/ ZODI_<Interface Id>_ <SrcSet>_<Context>.log	LKM - Runtime	SQL*Loader log file
<System Temp Dir or local FTP dir>/ ZODI_<Interface 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_<Interface Id>_ <SrcSet>_<Context>.err	LKM - Runtime	OS error output during SQL*Loader execution, may contain information, e.g. when SQL*Loader is not installed

7.7.5 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 column 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 8.1, "Introduction"](#)
- [Section 8.2, "Installation and Configuration"](#)
- [Section 8.3, "Defining the Topology"](#)
- [Section 8.4, "Setting up the Project"](#)
- [Section 8.5, "Creating and Reverse-Engineering a Model"](#)
- [Section 8.6, "Designing an Interface"](#)
- [Section 8.7, "Considerations for SAP ERP Integration"](#)

8.1 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 the *Oracle Data Integrator Getting Started with SAP ABAP ERP Adapter* on Technical Network (OTN):

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

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

8.1.1 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

8.1.2 Knowledge Modules

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

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

- Reads SAP data from SAP ERP system
- Loads this SAP data into Oracle or non-Oracle Staging Area
- Reverse-engineers SAP Metadata and proposes a tree browser to select only the required Metadata
- Uses flexfields to map the SAP table types (Transparent, Cluster, and Pool).

Table 8–1 SAP ERP 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.

8.1.3 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.

8.1.3.1 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.

8.1.3.2 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 interfaces 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 interface. 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 8.7.1, "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.

8.2 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](#)

8.2.1 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>.

8.2.2 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 8.2.1, "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 8.7.1, "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 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 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 interface (when this agent runs on the FTP server machine), or remotely (when this agent is located on a different machine than the FTP server). This FTP server must be accessible over the network from both the SAP 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 interfaces using LKM SAP ERP to Oracle (SQLLDR). SQL*Loader is used for loading data extracted from SAP to the Oracle staging area.

8.2.3 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](#)

8.2.3.1 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_misc` directory (Studio) and into `<ODI_HOME>/oracledi/agent/drivers` (Standalone Agent).
 5. Restart the ODI Components using SAP (ODI Studio, Standalone Agent)
 6. Check the JCo installation.

Note: Note: The SAP JCo software requires the `librfc32/librfc64` library, but does not include it. It is included e.g. with the SAP GUI software. In case this library is not yet installed, it has to be downloaded and installed following the steps outlined in [Appendix C, "Installing the librfc32/librfc64 Library."](#)

8.2.3.2 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.

8.2.3.3 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.

8.2.3.4 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.

8.2.4 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.

8.3 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](#)

8.3.1 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.

8.3.1.1 Create a File Data Server

Create a File data server as described in "Creating a File Data Server" of the *Oracle Fusion Middleware 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 *Oracle Fusion Middleware 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 8.7.1, "File Transfer Configurations"](#) for more information.

8.3.1.2 Create the File Schema

In this File data server create a Physical Schema as described in "Creating a File Physical Schema" of the *Oracle Fusion Middleware 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 8.7.1, "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`.

8.3.2 Create the SAP ABAP Data Server

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

8.3.2.1 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 *Oracle Fusion Middleware 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 8.7.2, "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 8.7.2, "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.

8.3.2.2 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 *Oracle Fusion Middleware 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 *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator* in the appropriate context.

8.4 Setting up the Project

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

Import the following KMs into your Oracle Data Integrator project:

- RKM 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 *Oracle Fusion Middleware Connectivity and Knowledge Modules Guide for Oracle Data Integrator* for a list of available KMs.

8.5 Creating and Reverse-Engineering a Model

This section contains the following topics:

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

8.5.1 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 *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.

8.5.2 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 *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*. This section details only the fields specific to SAP ERP:

1. In the Reverse 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 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 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.

8.6 Designing an Interface

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

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

2. Create joins, filters and mappings for your interface.
3. In the Flow tab of the interface, select the source set containing the SAP ABAP source table(s) and select the LKM SAP ERP to Oracle (SQLLDR)/LKM SAP ERP to SQL.

8.7 Considerations for SAP ERP Integration

This section includes the following topics:

- [File Transfer Configurations](#)
- [Controlling ABAP Uploading](#)
- [Log Files](#)
- [Limitations of the SAP ABAP Adapter](#)

8.7.1 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](#)

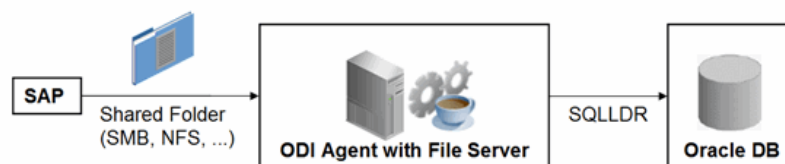
8.7.1.1 Transfer using a Shared Directory (recommended)

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 8–1 Configuration 1



This image shows the configuration where the shared directory is physically located on the ODI Agent machine.

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

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

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

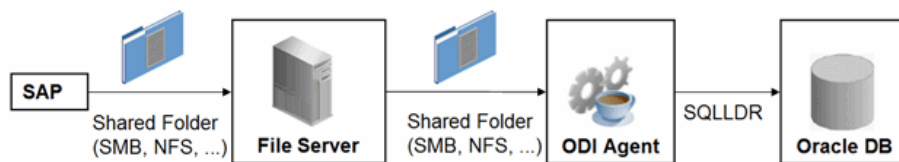
This configuration requires the following Topology settings:

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

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

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

Figure 8–2 Configuration 2



This image shows the configuration where the shared directory is not physically located on the ODI Agent machine.

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

1. The ABAP program extracts chunks of `FETCH_BATCH SIZE` records and writes them into a file in the shared folder.

2. The run-time agent copies the file into the directory given by TEMP_DIR option of the LKM.
3. for LKM SAP ERP to Oracle (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.

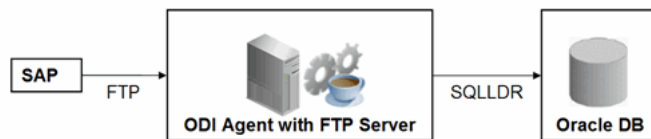
Please note that data files will be copied to the run-time agent from the shared folder in a local temporary folder. The default temporary directory is the system's temporary directory. On UNIX this is typically /tmp and on Windows c:\Documents and Settings\\Local Settings\Temp. This directory can be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

8.7.1.2 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 8–3 Configuration 1: FTP Server is installed on an ODI Agent machine



This image shows the file transfer configuration where the FTP server is installed on an ODI Agent machine.

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

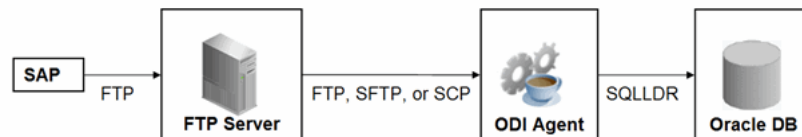
1. The ABAP program extracts the data and uploads the data file to the FTP server.
2. For LKM SAP ERP to Oracle (SQLLDR): SQL*Loader reads locally the data file and loads the data into the Oracle staging area.

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

This configuration requires the following Topology settings:

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

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



This image shows the file transfer configuration where the FTP server is not installed on an ODI Agent machine.

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

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

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

This configuration requires the following Topology settings:

1. Create a File data server pointing to the FTP server:
 - Host (Data Server): FTP server host name or IP address.

- User: User name to log into FTP server.
 - Password: Password for the user.
2. In this File data server create a physical schema representing the folder in the FTP host where the extraction file will be pushed. Specify the Data and Work Schemas as follows:
- *Directory (Schema)*: Path on the FTP server for uploading SAP extraction files.
 - *Directory (Work Schema)*: <undefined>; this path is left blank, as data files are never accessed directly from the FTP server's file system.

Considerations and Limitations:

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

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

8.7.2 Controlling ABAP Uploading

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

Once an Interface 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 *Oracle Fusion Middleware 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 an interface 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 interfaces
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 interface/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 Interface's ABAP extraction code and not to extract any data, such as for an automated installation. In this case all interfaces can be linked inside a package with the KM option `EXECUTE_ABAP_CODE` set to `False` in every interface. Executing this package will then install all ABAP code, but will not perform any execution.

To avoid the modification of all interfaces (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`.

8.7.3 Managing ODI SAP Transport Requests

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

An ODI interface 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 interface are assigned to one SAP function group. The ABAP extraction programs for a different interface will be assigned to a different SAP function group. The default function group name is similar to `ZODI_FGR_<Interface Id>`.

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

When the ABAP code of multiple ODI Interfaces should be grouped into a single transport request for more coarse-grained deployment control, the KM option `SAP_FUNCTION_GROUP_NAME` for all LKMs in these interfaces 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_<Interface Id>_<SourceSet Id>. This ensures convenient development due to unique program names. While the InterfaceId never changes, certain changes to an ODI interface 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.

8.7.4 Log Files

During the RKM and LKM execution several log files are created. These log files may contain valuable details for troubleshooting. Table 8–2 describes the different log files and their usage:

Table 8–2 Log Files

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

8.7.5 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 contains the following sections:

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

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

A.1 SAP ABAP BW Required Privileges

A 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, a SAP RFC user is sufficient.

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

- [Authorizations Required for RKM SAP BW Setup](#)
- [Authorizations Required for RKM Execution](#)
- [Authorizations Required for LKM Execution](#)
- [Authorizations Required for LKM Execution for Production](#)

A.1.1 Authorizations Required for RKM SAP BW Setup

The authorizations required for setting up the configuration for the RKM SAP BW and installing the required RFC programs to retrieve the metadata about SAP tables are the same as the authorizations required for the RKM SAP ERP.

See [Section B.1.1, "Authorizations Required for RKM SAP ERP Setup"](#) for more information.

A.1.2 Authorizations Required for RKM Execution

The authorizations required for running a reverse-engineering using the RKM SAP BW are the same as the authorizations required for the RKM SAP ERP.

See [Section B.1.2, "Authorizations Required for RKM Execution"](#) for more information.

A.1.3 Authorizations Required for LKM Execution

The authorizations required for using the LKM SAP BW to Oracle (SQLDR) are the same as the authorizations required for the LKM SAP ERP to Oracle (SQLDR).

See [Section B.1.3, "Authorizations Required for LKM Execution"](#) for more information.

A.1.4 Authorizations Required for LKM Execution for Production

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

See [Section B.1.4, "Authorizations Required for LKM Execution for Production"](#) for more information.

A.2 SAP Stand-Alone Connection Test

In addition to the Connection Testing described in the *Getting Started with SAP ABAP Adapter guide*, 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.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_misc 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_misc directory.
5. Launch the utility using the following command:

```
java -cp sapjco3.jar;../oracledi/agent/lib/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_misc folder.

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 = Yes).

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

A.4 Uninstalling ODI SAP Components

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

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

To 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 ZODIBW_PCKG in the package or development class field.
6. Right-click the ZODIBW_PCKG object below the object name tab.
7. Select **Delete** in the drop down menu and delete the development class.

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

8. 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.

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 contains the following sections:

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

B.1 SAP ABAP ERP Required Privileges

A 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, a SAP RFC user is sufficient.

The following sections list the privileges required for using SAP ERP Knowledge Modules:

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

B.1.1 Authorizations Required for RKM SAP ERP Setup

The authorizations listed in [Table B-1](#) are required for setting up the configuration for the RKM SAP ERP and install the required RFC programs to retrieve the metadata about SAP tables.

Table B-1 RKM SAP ERP Setup Authorizations

Object	Field Name	Value
S_ADMI_FCD	S_ADMI_FCD	*
S_ADMI_FCD	S_ADMI_FCD	NADM
S_ADMI_FCD	S_ADMI_FCD	PADM
S_CTS_ADMI	CTS_ADMFCT	EPS1
S_CTS_ADMI	CTS_ADMFCT	EPS2
S_CTS_ADMI	CTS_ADMFCT	PROJ
S_C_FUNCT	ACTVT	16
S_C_FUNCT	CFUNCNAME	*
S_C_FUNCT	PROGRAM	*
S_CTS_ADMI	CTS_ADMFCT	TABL
S_TCODE	TCD	SU53
S_TCODE	TCD	SE38
S_TCODE	TCD	SE80
S_DATASET	ACTVT	6
S_DATASET	ACTVT	33
S_DATASET	ACTVT	34
S_DATASET	FILENAME	*
S_TCODE	TCD	SE91
S_TCODE	TCD	ST22
S_DATASET	PROGRAM	*
S_RFC	RFC_TYPE	*
S_RFC	RFC_NAME	*
S_RFC	ACTVT	16
S_DEVELOP	ACTVT	*
S_DEVELOP	DEVCLASS	*
S_DEVELOP	OBJNAME	*
S_DEVELOP	OBJTYPE	*
S_DEVELOP	P_GROUP	*
S_DOKU_AUT	DOKU_ACT	RAW_VERS
S_DOKU_AUT	DOKU_DEVCL	TEST
S_DOKU_AUT	DOKU_MODE	MAINTAIN
S_RFC_ADM	ACTVT	3
S_RFC_ADM	ICF_VALUE	*
S_RFC_ADM	RFCDEST	*
S_RFC_ADM	RFCTYPE	*
S_RZL_ADM	ACTVT	*

Table B-1 (Cont.) RKM SAP ERP Setup Authorizations

Object	Field Name	Value
S_TABU_DIS	ACTVT	*
S_TABU_DIS	DICBERCLS	*
S_TCODE	TCD	AL11
S_TCODE	TCD	SE10
S_TCODE	TCD	SE11
S_TCODE	TCD	SE16
S_TCODE	TCD	SE37
S_TCODE	TCD	SM58
S_TCODE	TCD	SM59
S_TRANSPRT	ACTVT	1
S_TRANSPRT	ACTVT	2
S_TRANSPRT	ACTVT	3
S_TRANSPRT	ACTVT	5
S_TRANSPRT	ACTVT	6
S_TRANSPRT	ACTVT	23
S_TRANSPRT	ACTVT	43
S_TRANSPRT	ACTVT	65
S_TRANSPRT	ACTVT	78
S_TRANSPRT	ACTVT	90
S_TRANSPRT	TTYPE	CUST
S_TRANSPRT	TTYPE	DTRA
S_TRANSPRT	TTYPE	MOVE
S_TRANSPRT	TTYPE	PIEC
S_TRANSPRT	TTYPE	TASK
S_TRANSPRT	TTYPE	TRAN

Table B-2 lists the transaction codes.

Table B-2 Transaction Codes

SU53
SE38
SE80
SE91
ST22
AL11
SE10
SE11

Table B–2 (Cont.) Transaction Codes

SE16
SE37
SM58
SM59

B.1.2 Authorizations Required for RKM Execution

The authorizations listed in [Table B–3](#) are required for running a reverse-engineering using the RKM SAP ERP.

Table B–3 Authorizations for RKM Execution

Object	Field Name	Value
S_ADMI_FCD	S_ADMI_FCD	*
S_ADMI_FCD	S_ADMI_FCD	NADM
S_ADMI_FCD	S_ADMI_FCD	PADM
S_CTS_ADMI	CTS_ADMFCT	EPS1
S_CTS_ADMI	CTS_ADMFCT	EPS2
S_CTS_ADMI	CTS_ADMFCT	PROJ
S_C_FUNCT	ACTVT	16
S_C_FUNCT	CFUNCNAME	*
S_C_FUNCT	PROGRAM	*
S_CTS_ADMI	CTS_ADMFCT	TABL
S_TCODE	TCD	SU53
S_TCODE	TCD	SE38
S_TCODE	TCD	SE80
S_DEVELOP	ACTVT	16
S_DATASET	ACTVT	33
S_DATASET	FILENAME	*
S_TCODE	TCD	SE91
S_TCODE	TCD	ST22
S_DATASET	PROGRAM	*
S_RFC	RFC_TYPE	*
S_RFC	RFC_NAME	*
S_RFC	ACTVT	16
S_DEVELOP	ACTVT	3
S_DEVELOP	DEVCLASS	*
S_DEVELOP	OBJNAME	*
S_DEVELOP	OBJTYPE	*
S_DEVELOP	P_GROUP	*

Table B-3 (Cont.) Authorizations for RKM Execution

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

B.1.3 Authorizations Required for LKM Execution

The authorizations listed in [Table B-4](#) are required for using the LKM SAP ERP to Oracle (SQLDR). This set of authorization is needed to install the ABAP code generated by the LKM and execute it.

Table B-4 Authorizations for LKM Execution

Object	Field Name	Value
S_ADMI_FCD	S_ADMI_FCD	*
S_ADMI_FCD	S_ADMI_FCD	NADM
S_ADMI_FCD	S_ADMI_FCD	PADM
S_CTS_ADMI	CTS_ADMFCT	EPS1
S_CTS_ADMI	CTS_ADMFCT	EPS2

Table B-4 (Cont.) Authorizations for LKM Execution

Object	Field Name	Value
S_CTS_ADMI	CTS_ADMFCT	PROJ
S_C_FUNCT	ACTVT	16
S_C_FUNCT	CFUNCNAME	*
S_C_FUNCT	PROGRAM	*
S_CTS_ADMI	CTS_ADMFCT	TABL
S_TCODE	TCD	SU53
S_TCODE	TCD	SE38
S_TCODE	TCD	SE80
S_DEVELOP	ACTVT	16
S_DATASET	ACTVT	33
S_DEVELOP	ACTVT	7
S_DATASET	FILENAME	*
S_DEVELOP	ACTVT	1
S_TCODE	TCD	SE91
S_TCODE	TCD	ST22
S_DEVELOP	ACTVT	40
S_DEVELOP	ACTVT	41
S_DATASET	ACTVT	34
S_DEVELOP	ACTVT	94
S_DEVELOP	ACTVT	2
S_TRANSPRT	ACTVT	2
S_TRANSPRT	ACTVT	1
S_TRANSPRT	ACTVT	60
S_TABU_DIS	ACTVT	2
S_RZL_ADM	ACTVT	1
S_DATASET	PROGRAM	*
S_RFC	RFC_TYPE	*
S_RFC	RFC_NAME	*
S_DEVELOP	ACTVT	*
S_DEVELOP	DEVCLASS	*
S_DEVELOP	OBJNAME	*
S_DEVELOP	OBJTYPE	*
S_DEVELOP	P_GROUP	*
S_DOKU_AUT	DOKU_ACT	RAW_VERS
S_DOKU_AUT	DOKU_DEVCL	TEST
S_DOKU_AUT	DOKU_MODE	MAINTAIN

Table B–4 (Cont.) Authorizations for LKM Execution

Object	Field Name	Value
S_RFC_ADM	ACTVT	3
S_RFC_ADM	ICF_VALUE	*
S_RFC_ADM	RFCDEST	*
S_RFC_ADM	RFCTYPE	*
S_RZL_ADM	ACTVT	3
S_TABU_DIS	ACTVT	3
S_TABU_DIS	DICBERCLS	*
S_TCODE	TCD	AL11
S_TCODE	TCD	SE10
S_TCODE	TCD	SE11
S_TCODE	TCD	SE16
S_TCODE	TCD	SE37
S_TCODE	TCD	SM58
S_TCODE	TCD	SM59
S_TRANSPRT	ACTVT	3
S_TRANSPRT	TTYPER	CUST
S_TRANSPRT	TTYPER	DTRA
S_TRANSPRT	TTYPER	MOVE
S_TRANSPRT	TTYPER	PIEC
S_TRANSPRT	TTYPER	TASK
S_TRANSPRT	TTYPER	TRAN

B.1.4 Authorizations Required for LKM Execution for Production

The authorizations listed in [Table B–5](#) are required for using the LKM SAP ERP to Oracle (SQLLDR) in a production environment. In such environment, the ABAP code generated by the LKM is not installed by the LKM but simply executed.

Table B–5 Authorizations for LKM Execution for Production

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

Table B-5 (Cont.) Authorizations for LKM Execution for Production

Object	Field Name	Value
S_C_FUNCT	PROGRAM	*
S_CTS_ADMI	CTS_ADMFCT	TABL
S_TCODE	TCD	SU53
S_TCODE	TCD	SE38
S_TCODE	TCD	SE80
S_DEVELOP	ACTVT	16
S_DATASET	ACTVT	33
S_DATASET	FILENAME	*
S_TCODE	TCD	SE91
S_TCODE	TCD	ST22
S_DATASET	PROGRAM	*
S_RFC	RFC_TYPE	*
S_RFC	RFC_NAME	*
S_RFC	ACTVT	16
S_DEVELOP	ACTVT	3
S_DEVELOP	DEVCLASS	*
S_DEVELOP	OBJNAMEv	*
S_DEVELOP	OBJTYPE	*
S_DEVELOP	P_GROUP	*
S_DOKU_AUT	DOKU_ACT	RAW_VERS
S_DOKU_AUT	DOKU_DEVCL	TEST
S_DOKU_AUT	DOKU_MODE	MAINTAIN
S_RFC_ADM	ACTVT	3
S_RFC_ADM	ICF_VALUE	*
S_RFC_ADM	RFCDEST	*
S_RFC_ADM	RFCATYPE	*
S_RZL_ADM	ACTVT	3
S_TABU_DIS	ACTVT	3
S_TABU_DIS	DICBERCLS	*
S_TCODE	TCD	AL11
S_TCODE	TCD	SE10
S_TCODE	TCD	SE11
S_TCODE	TCD	SE16
S_TCODE	TCD	SE37
S_TCODE	TCD	SM58
S_TCODE	TCD	SM59

Table B-5 (Cont.) Authorizations for LKM Execution for Production

Object	Field Name	Value
S_TRANSPRT	ACTVT	3
S_TRANSPRT	TTYPE	CUST
S_TRANSPRT	TTYPE	DTRA
S_TRANSPRT	TTYPE	MOVE
S_TRANSPRT	TTYPE	PIEC
S_TRANSPRT	TTYPE	TASK
S_TRANSPRT	TTYPE	TRAN

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 *Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator*.
2. In the Reverse 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_misc` 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_misc` directory.
5. Launch the utility using the following command:

```
java -cp sapjco3.jar;../oracledi/agent/lib/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_misc folder.

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 = Yes).

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

B.4 Uninstalling ODI SAP Components

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

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

To 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_DEVCLASS in the package or development class field.
6. Right-click the ZODI_DEVCLASS 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.

Installing the librfc32/librfc64 Library

This appendix describes how to install the librfc32 or librfc64 library.

The SAP JCo software requires the librfc32/librfc64 library, but does not include it. It is included for example with the SAP GUI software. In case this library is not yet installed, this section describes how to download and install it. These steps are only required, if librfc32/librfc64 is not yet installed.

C.1 Installing the librfc32/librfc64 Library

Perform the following steps to install the librfc32/librfc64 library:

1. Connect to SAP Support Portal <http://service.sap.com>.
2. Go to SAP support portal.
3. Go to Software Downloads.
4. Go to the SAP Software Download Center.
5. Select Support Packages and Patches.
6. Select Browse Our Download Catalog.
7. Select Additional Components.
8. Download appropriate RFCSDK version (SAP NW RFC SDK, SAP RFC SDK, SAP RFC SDK UNICODE)
9. Use SAPCAR extractor to extract the content of the downloaded SAR file
10. For JCo installed on Windows:

Copy the file `librfc[u][32|64].dll` located in folder `rfcsdk/bin` to `c:\windows\system32` (exact folder name depends on windows version).

For JCo installed on Unix:

Copy the file `librfccm[u].so` to the folder containing the `sapjco3.jar` and `libsapjco3.so`. This directory may need to be added to `LD_LIBRARY_PATH`.

