

SILWOOD TECHNOLOGY LTD

---

Saphir Metadata Explorer

# Getting Started Guide

SILWOOD TECHNOLOGY LIMITED

# Getting Started Guide – Saphir 5.5

---

This product is subject to the license agreement and limited warranty accompanying the product. The product software may be used or copied only in accordance with the terms of this agreement.

Information in this document is subject to change without notice. No part of this manual may be reproduced, or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Silwood Technology Ltd, or Authorised Silwood Technology Distributor.

© Copyright 2005-11 Silwood Technology Ltd. All rights reserved.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	How does Saphir work? .....	4
1.2	Which Enterprise Applications and Versions does Saphir address? .....	4
1.3	Saphir manual set.....	4
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Saphir setup tasks .....	5
2.2	Saphir system and resource requirements.....	5
2.2.1	System requirements .....	5
2.2.2	Resource requirements .....	6
2.2.3	How Saphir communicates with databases .....	6
2.3	Creating the Saphir database.....	6
2.3.1	To create the Saphir database.....	7
2.3.2	Database user requirements.....	7
2.4	Installing Saphir .....	7
2.5	Starting Saphir for the first time.....	7
2.6	Using the Repository Manager to configure Saphir .....	8
2.6.1	To Configure a new Saphir repository .....	9
2.7	Repository Maintenance.....	16
2.8	Configuring Saphir to connect to an SAP system .....	17
2.8.1	Connecting to SAP using the Direct Connection Method .....	17
2.8.2	Connecting to SAP using the ABAP Method.....	17
2.8.3	Required Access Rights for the SAP User used by Saphir.....	18
2.8.4	Entering the extraction method choice into Saphir .....	18
2.8.5	Entering the SAP database connection settings .....	19
2.9	Entering the SAP RFC connection settings .....	24
2.10	Connecting to PeopleSoft Enterprise .....	25
2.10.1	Choosing the PeopleSoft database type.....	25
2.10.2	Entering the PeopleSoft database connection settings.....	26
2.11	Connecting to J.D.Edwards EnterpriseOne .....	30
2.11.1	To produce the XML Files from the EnterpriseOne system .....	30
2.11.2	Choosing the EnterpriseOne database type.....	31
2.11.3	Entering the EnterpriseOne database connection settings.....	32
2.12	Connecting to Siebel.....	36
2.12.1	Choosing the Siebel database type.....	36
2.12.2	Entering the Siebel database connection settings.....	37
2.13	Connecting to Oracle Enterprise Business Suite.....	42
2.13.1	Choosing the EBS database type .....	42
2.13.2	Entering the EBS database connection settings .....	43
2.14	Opening a Saphir Repository.....	46
2.14.1	To Open an existing Repository .....	46
<b>3</b>	<b>Extracting Metadata</b>	<b>47</b>
3.1	Extracting metadata from SAP or SAP BW.....	47
3.1.1	Why would I use the Export/Import options? .....	50

---

3.2	Extracting metadata from PeopleSoft Enterprise .....	53
3.3	Extracting metadata from Siebel.....	57
3.4	Extracting metadata from J.D.Edwards EnterpriseOne .....	60
3.5	Extracting metadata from Oracle EBS .....	64
3.6	Restarting an extraction.....	67
<b>4</b>	<b>Quick Tour of Saphir's Features</b>	<b>68</b>
4.1	Model Overview .....	68
4.1.1	Tables.....	68
4.1.2	Fields.....	70
4.1.3	Table Relationships.....	71
4.1.4	Filtering / Searching.....	71
4.1.5	Subject Areas.....	72
4.1.6	Filtering using Subject Areas .....	73
4.1.7	Expanding tables in a Subject Area .....	73
4.1.8	Advanced Search.....	73
4.2	Navigating tables via the Application Hierarchy .....	74
4.3	Exporting to third party tools.....	74
<b>Appendix A</b>	<b>Installing the SAP ABAP Function</b>	<b>76</b>
A.1	Create a Package.....	76
A.2	Create a Function Group .....	77
A.3	Create the ABAP Function Module .....	78
A.4	Testing the Function.....	81
<b>Appendix B</b>	<b>The SaphirSettings.xls file</b>	<b>83</b>
B.1	File usage .....	83
B.2	File structure .....	83
<b>Appendix C</b>	<b>Using SAPRFC.ini with Saphir</b>	<b>87</b>
C.1	Creating the SAPRFC.ini file.....	87
C.2	Using SAPRFC.ini from Saphir .....	88
<b>Appendix D</b>	<b>Connecting to a SQLAnywhere database</b>	<b>90</b>
D.1	Settings for connecting to the database .....	90
<b>Appendix E</b>	<b>Using the ABAP 'ALL' Functions</b>	<b>92</b>
E.1	Installing the 'All' Functions.....	92
E.2	Testing the Function.....	93
<b>Index</b>		<b>95</b>

---

# 1 Introduction

*An overview of the Saphir product and the Enterprise Applications it supports*

**S**aphir, is a metadata exploration tool for enterprise applications like SAP, PeopleSoft, Siebel and Oracle E-Business Suite. Saphir makes the data definitions from these major ERP (Enterprise Resource Planning) packages available in an easy to understand form, allowing users to explore ERP data structures without specialist application knowledge.

This manual describes how to install Saphir, how to extract and manipulate the ERP metadata, and gives an overview of the main product features.

Please refer to the '*Saphir User Guide*' for full details on all product features.

## 1.1 How does Saphir work?

Using the Saphir product involves two distinct stages; the extraction of metadata from the chosen environment; the exploration of that extracted metadata. The metadata extraction stage is performed by attaching to the 'source' package (e.g. SAP) where the required metadata is stored. The exact method of achieving this is dependent on the source product. The following sections describe this process in detail. The extracted metadata is used to populate the Saphir Repository. This is a set of relational tables where the various extracted objects (e.g. tables, relationships, indexes...) are stored. A range of popular RDBMS' can be used for the Repository. Once the metadata is extracted, the second stage of Saphir usage comes into effect: the browsing of the metadata.

The browsing stage is the real value of the Saphir environment. The user interface enables users to search for required data items like tables or columns, explore relationships between tables and create subsets of the metadata for later review, or optionally, for export into a range of tool environments.

## 1.2 Which Enterprise Applications and Versions does Saphir address?

Saphir supports the following Enterprise Applications:

- SAP version 3.1 and above
- SAP Business Warehouse (BW) version 2.2 and above.
- PeopleSoft Enterprise applications from Version 8 and above
- J.D.Edwards EnterpriseOne Xe and above
- Siebel version 6.5 and above
- Oracle E-Business Suite version 11 and above

## 1.3 Saphir manual set

There are two Saphir manuals supplied in Adobe Acrobat format (.pdf files) with the Saphir software.

- Getting Started Guide (this manual): Use this manual for details of product installation and an overview of Saphir features
- User Guide: Describes all of the Saphir functions in detail

## 2 Installation

*How to install and configure the Saphir software, ready for metadata extraction*

The Saphir installation process requires a number of steps to be performed, in addition to the actual ‘setup’ of the Saphir product itself. The following sections describe these steps in detail. Please take the time to read and understand this chapter before embarking on the installation.

### 2.1 Saphir setup tasks

- Check system and resource requirements. Ensure that your system meets the minimum requirements necessary to install and run Saphir. There may also be additional personnel and hardware/software resources required that you need to assemble before starting the installation. See ‘Saphir System Requirements’ below for more information.
- Create a database for storing the Saphir metadata. Saphir uses a range of RDBMS types to instantiate a database where the extracted metadata is stored. See ‘Creating the Saphir Database’ below for more information.
- Install Saphir. See ‘Installing Saphir’ below for more information.
- Configure the software. There are a series of configuration tasks that must be performed to allow Saphir to talk to the Saphir repository. See ‘Using the Repository Manager to configure Saphir’ below.
- Connect to the enterprise application. The final setup task is to connect to the Enterprise Application where the metadata is stored and perform an extraction. See ‘Connecting to your Enterprise Application’ below.

### 2.2 Saphir system and resource requirements

Before installing, ensure that you have the necessary system and resource requirements, as detailed below. It is also useful to understand how Saphir communicates with the RDBMS. See ‘How Saphir communicates with databases’ below for more information on this, including the required software components.

#### 2.2.1 System requirements

The PC on which Saphir is to be installed should have the following:

- A minimum of 1 Gigabytes of RAM
- 40 Megabytes of hard disk space for the Saphir Software
- The appropriate client connectivity software (e.g. Oracle Net 9) for connection to the Saphir Repository database and (unless ABAP extraction is to be used for SAP) the Enterprise Application database. See also the section ‘How Saphir communicates with databases’ below.
- Where Saphir is being used with SAP or SAP BW and the user want to have control of running the extraction of metadata from the SAP system, then the SAP GUI client software needs to be installed on the PC, if the metadata is to be extracted from SAP via RFC/ABAP. (Note: The ABAP functions provided with Saphir can be run independently of the product. In this case, SAP GUI would not be required on the PC. See (Appendix E: Using the ABAP ‘ALL’ Functions for more details.)

### 2.2.2 Resource requirements

A full Saphir installation may require the involvement of a range of personnel and resources across the organisation. This section lists the main resources you should have in place before installing Saphir

- A database for storing the extracted metadata. You may need to contact a Database Administrator (DBA) to create a new database for the Saphir Repository, and to grant appropriate access rights to that database. Saphir can also use a SQLite database. SQLite is an open source RDBMS that is now embedded in the Saphir software.
- Access to the Enterprise Application. Saphir needs access to the Enterprise Application in order to extract the metadata stored in its data dictionary. The actual method for extraction is Application dependent, but you will need to gain access to the system before an extraction of metadata can be performed.

### 2.2.3 How Saphir communicates with databases

A database connection will be required for each database with which Saphir needs to communicate. This will normally be two connections for each 'Enterprise Application' to be reverse engineered:

- One for the Saphir Repository
- One for the 'source' Enterprise Application database

The only exception to this is where Saphir is being used with an SAP system and the connection to the SAP system is via ABAP. In this case, there is no need for a connection to the SAP database, as the extraction of metadata is performed by the ABAP function, not by a SQL connection.

Saphir uses Microsoft ActiveX Data Objects (ADO) to communicate with databases. (Note: this does not apply to the SQLite database type).

The ADO technology is part of the Microsoft Data Access Components (MDAC). It is recommended that MDAC 2.8 or higher is installed on each PC where Saphir is to be installed. The table below shows the recommended ADO Providers and versions that are required for each of the RDBMS types Saphir supports.

Database	ADO Provider
Oracle	Oracle Provider for OLE DB - Version 8.1.7 or later
Microsoft SQL Server	Microsoft OLE DB Provider for SQL Server - Version 8.1 or later
IBM DB2	IBM OLEDB Provider for DB2
Sybase SQL Anywhere	SQL Anywhere OLEDB Provider

## 2.3 Creating the Saphir database

The Saphir Repository is a set of database tables for storing the extracted SAP, PeopleSoft Enterprise, J.D.Edwards EnterpriseOne (formerly J.D. Edwards OneWorld), Oracle E-Business Suite or Siebel metadata. It will be necessary to create a new database to store this information.

The exact sequence of steps for setting up the Saphir Repository is specific to the target RDBMS and will require some knowledge of that software's capabilities.

### 2.3.1 To create the Saphir database

1. Decide on the target RDBMS in which the Saphir repository is to be established. This can be Oracle, Microsoft SQL Server (version 7 or later), IBM DB2 UDB, SQL Anywhere or SQLite.
2. Create a new database in the chosen RDBMS.

**For Oracle:** The database should be a minimum of 400MB. One large rollback segment of 50MB is needed. Other Rollback segments can use Oracle defaults. Use default Oracle extents.

**For SQL Server:** Place the database and Transaction Log on different physical devices. Make the database 400MB and the Transaction Log 50MB.

**For DB2:** Create a new database for the Saphir Repository. The User space should be 400Mb and the Temp space 50Mb. A temporary system tablespace with 8k Blocks has to exist in order to allow Saphir to run properly.

**For SQL Anywhere:** Create a new database for the Saphir Repository.

**For SQLite:** Saphir can create this database type as part of the repository configuration process.

Note: Whilst Saphir will work well with the SQLite database, we do not recommend this platform other than for personal usage of the product or as a quick route to evaluating Saphir.

### 2.3.2 Database user requirements

A Database User is required for the Saphir database. This must have Table Owner permissions so that the various objects (tables, constraints, etc.) that are required in the Saphir database can be created when the Saphir repository setup process is run.

Where more than one Saphir license is to be used to access the same Saphir Repository, each Saphir license in this group must use the same Database User (i.e. the Table Owner).

## 2.4 Installing Saphir

The Saphir Setup program controls the installation process.

To install Saphir locate the Saphir setup file and run this program.

Follow the instructions of the Saphir Setup program through installation.

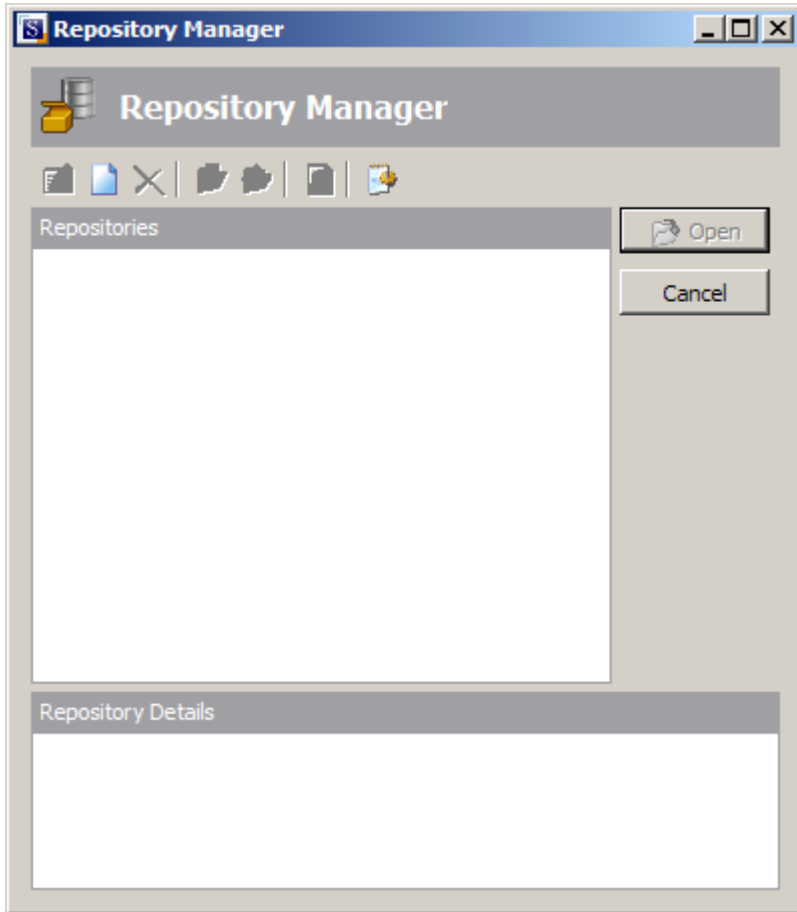
## 2.5 Starting Saphir for the first time

The first time Saphir is run you may be prompted to provide licensing details. The exact screen displayed will depend upon which Saphir version you have purchased.

Saphir then displays the Repository Manager.


## 2.6 Using the Repository Manager to configure Saphir

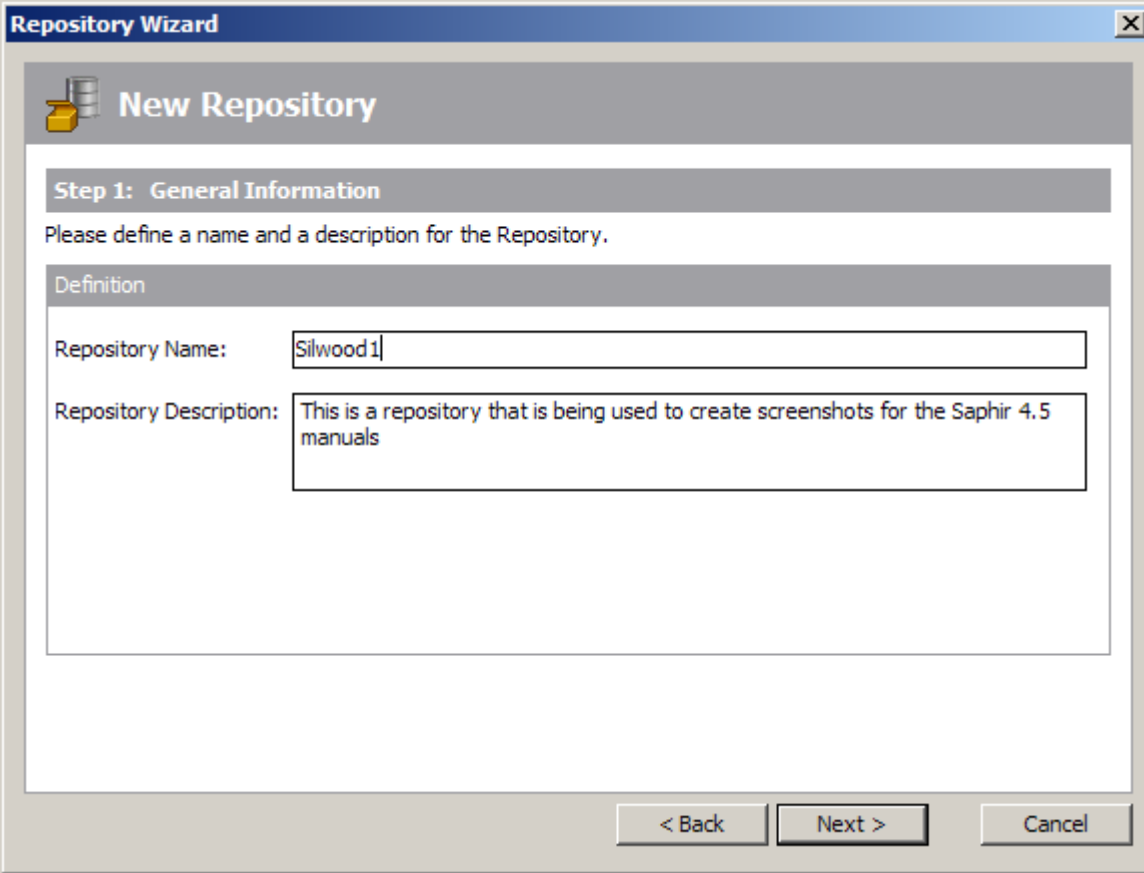
The Repository Manager uses a Wizard to guide you through the setup steps for configuring Saphir and connecting to your chosen Enterprise Application.



The Repository Manager

## 2.6.1 To Configure a new Saphir repository

From the Repository Manager, click the  button to start the process of configuring a new repository. Click the 'Next' button on the first 'Repository Wizard' form to display the 'Step 1' form.



The screenshot shows a dialog box titled "Repository Wizard" with a close button (X) in the top right corner. The main title is "New Repository" with a small icon of a yellow box and a silver cylinder. Below this is a section titled "Step 1: General Information" with a grey background. The text says "Please define a name and a description for the Repository." Below this is a section titled "Definition" with a grey background. There are two input fields: "Repository Name:" with the text "Silwood1" and "Repository Description:" with the text "This is a repository that is being used to create screenshots for the Saphir 4.5 manuals". At the bottom of the dialog box are three buttons: "< Back", "Next >", and "Cancel".

Repository Wizard – Step 1 – General Information

Specify a name in the Repository Name field, and optionally a Repository Description. Click the Next button to display the Step 2 form.

**Repository Wizard**

**New Repository**

**Step 2: Database Configuration**

Please define the database type and the connection information of the repository.

**Database Type**

Oracle
  MS SQL
  SQLite
  SQL Anywhere
  DB2

**Database Connection**

Connection Name:  New Edit  
 Database Username:   
 Database Password:   
 Character Encoding:

Repository Wizard – Step 2 – Database Configuration

Select the Radio Button that corresponds to the Database Type being used to host the Saphir Repository. The Database Connection properties in the lower half of the screen will change depending on the Database Type selected.

For a SQLite database, the following Database Connection properties are displayed:

**Database Connection SQLite**

DataBase File Name:  ...

Cache Size DB Engine:  [MB]

SQLite Database Connection properties

Specify a location for the SQLite database file in the DataBase file name field. Then click the 'Create Database' button to create the database. Note that this creates not only the database but the Saphir repository tables within that database.

Click the 'Next' button to display the Repository Wizard Step 3 form (see below).

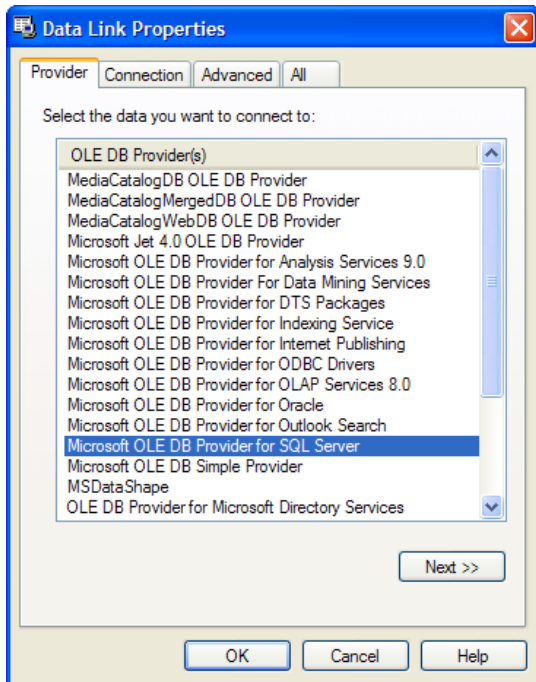
For all other database types, the following Database Connection properties are displayed:

The screenshot shows a 'Database Connection' dialog box with the following fields and controls:

- Connection Name:** A text box followed by a dropdown arrow, with 'New' and 'Edit' buttons to its right.
- Database Username:** A text box.
- Database Password:** A text box.
- Character Encoding:** A dropdown menu currently showing 'ANSI'.
- Command Timeout:** A text box followed by '[sec]'.
- Test Connection:** A button at the bottom left.

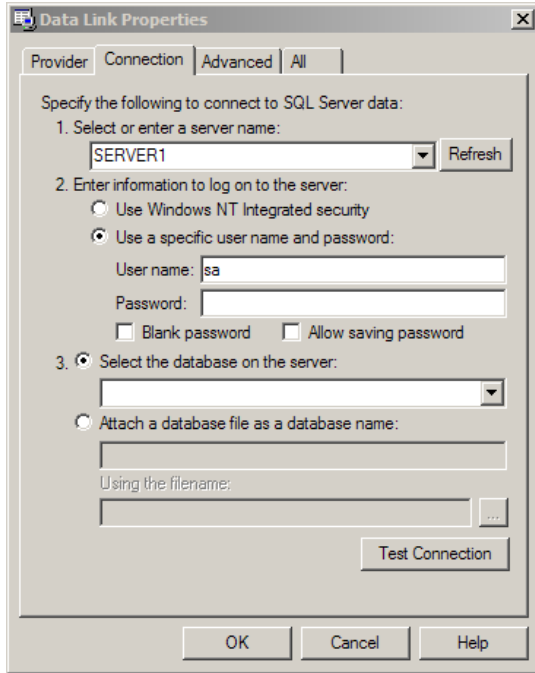
Database Connection properties for RDBMS types other than SQLite

1. The Database Alias dropdown list shows all the existing ADO connections previously configured. If this is a new Saphir installation, it will be necessary to create a new connection. Click on the 'New' button to the right of the drop down to begin this process.
2. The 'Edit Connection Definition' form is now displayed. Enter an appropriate name for the connection in the 'Connection Definition name' field and click the 'OK' button. The ADO Data Link Properties form is then displayed.

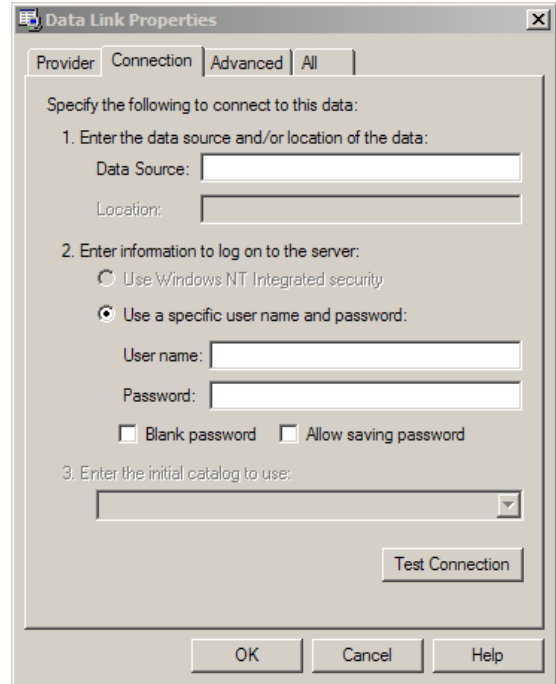


ADO Data Link Properties Form

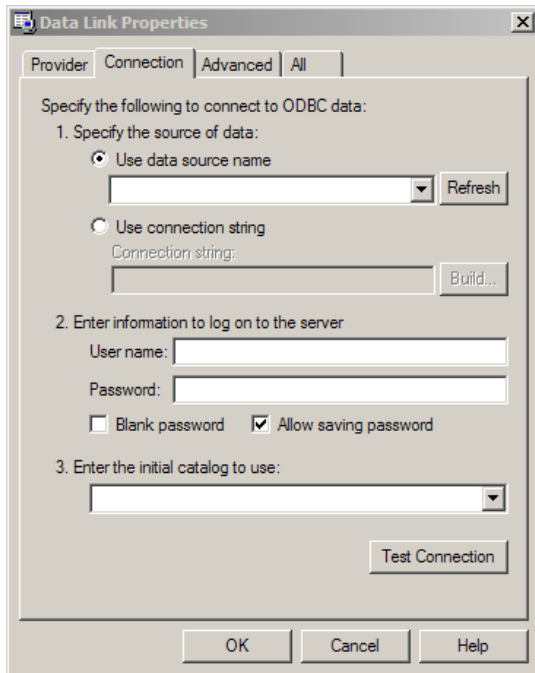
3. If the database connection being created is Oracle, select 'Oracle Provider for OLE DB' from the list of Providers. If the database connection being created is SQL Server, select 'Microsoft OLE DB Provider for SQL Server'. For DB2 select the 'IBM OLEDB Provider for DB2', and for Sybase SQL Anywhere select 'SQL Anywhere OLEDB Provider'. Now click the 'Next' button.
4. The Data Link Properties Connection tab is now displayed. The available fields on this tab will be dependent on the chosen Provider. The layouts for SQL Server, Oracle, DB2 and SQL Anywhere are shown below.



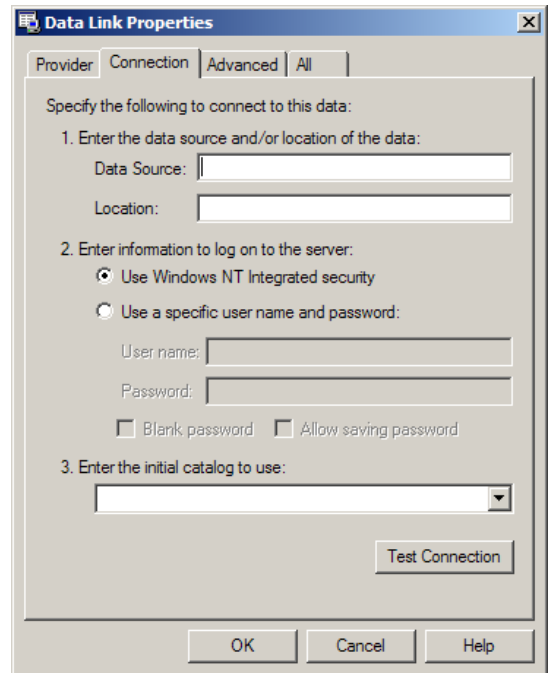
Data Link Properties 'Connection' tab for SQL Server



...for Oracle...

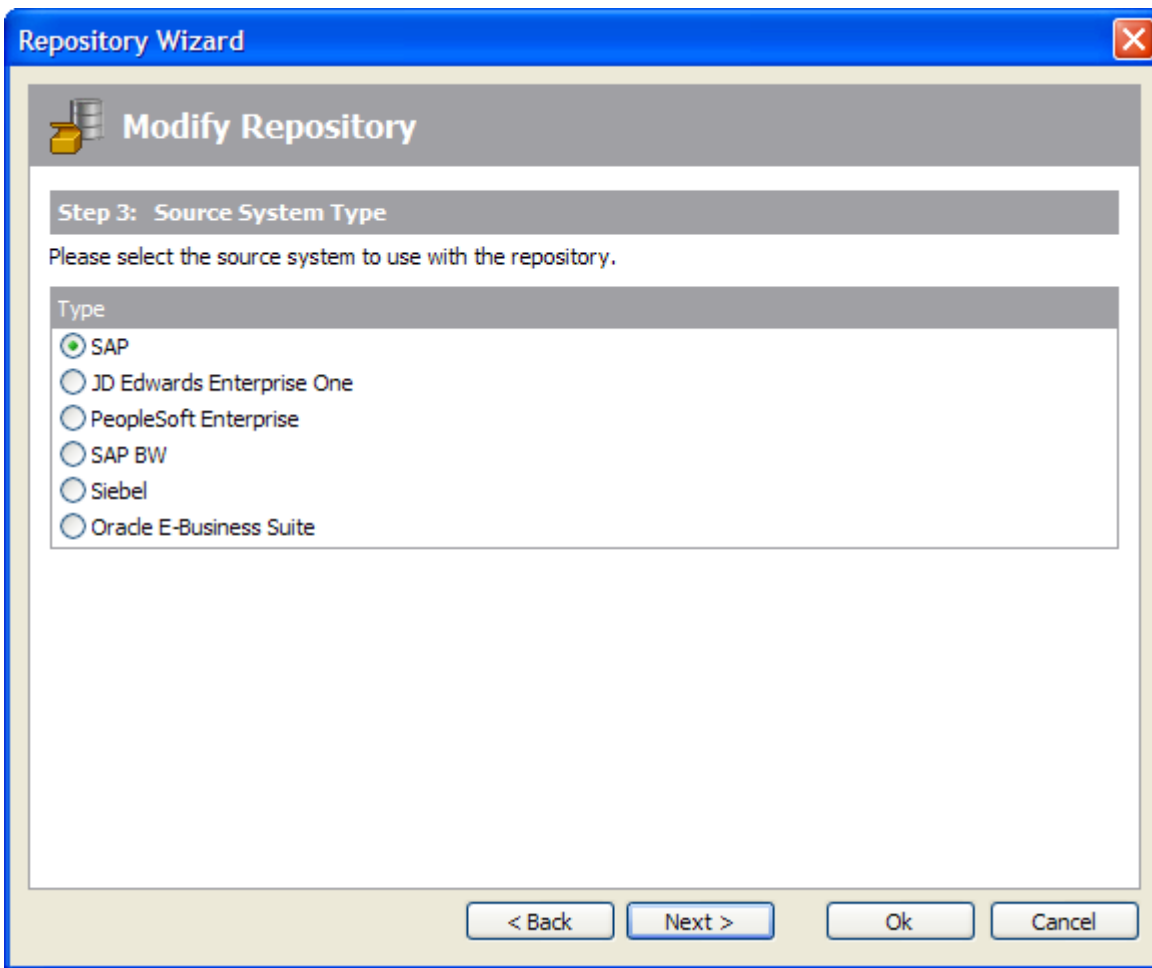


...for DB2....



...and for SQL Anywhere

5. For an Oracle database, in the 'Data Source' field, enter the connection string for connecting to the Oracle database. This will normally be the string used in the TNSNAMES file for connection to the database. For a SQL Server database, in the 'Select or enter a server name' field, enter or select the name of the Server where the database is located. If the database is DB2, enter the database name in the 'Data Source'. For SQL Anywhere, a connection string needs to be specified in order to connect to the database. See Appendix D for details of this connection string.
6. Enter a User name and Password. The User Name is a valid database User-id with owner access rights to the Saphir Repository. Alternatively, leave these fields blank and enter the User Name and Password in the Repository Wizard form.
7. If a User name and Password were entered in the previous step, test the connection to the database by clicking the 'Test Connection' button.. If the connection is successful, click 'OK' to return to the Repository Wizard.
8. If not previously entered in the ADO configuration, enter the owner username and password for the database
9. Select the Character Encoding for the database. This can be ANSI, UTF-8 or UTF-16.
10. For Microsoft SQL Server and DB2, a 'Command Timeout' property may be specified. This indicates the length of time to allow for a given query to complete. The default is 210 seconds. It will not normally be necessary to enter a value for this field unless 'timeout' problems are encountered.
11. Click the 'Text Connection' button to ensure that a connection can be established between Saphir and the specified database.
12. Click the 'Next' button to display the Repository Wizard Step 3 form.



Repository Wizard – Step 3 – Source System Type

Step 3 is about specifying the type of Enterprise Application that Saphir will connect to. Check the radio button that corresponds to the 'source' system that this Repository is to be used with.

Note: The process of specifying a source system is specific to each Repository. A number of ERP instances (e.g. Production, Development) and/or Systems (e.g. SAP, PeopleSoft) may be managed by the same Saphir installation. The ability to manage multiple ERPs is controlled by the type of Saphir license your organisation has purchased.

The next steps in the Repository Wizard are about specifying the location and connection method for the 'source' Enterprise Application.. These steps are specific to each Enterprise Application. You can either carry on with the Wizard steps now, or click through each screen and return to specify the specific connection parameters later.

To connect to an SAP or SAP BW system, see 'Configuring Saphir to connect to an SAP system' below


For a PeopleSoft Enterprise system, see 'Connecting to PeopleSoft Enterprise' below

For a J.D.Edwards EnterpriseOne system, see 'Connecting to J.D.Edwards EnterpriseOne' below.


For a Siebel system, see 'Connecting to Siebel' below.

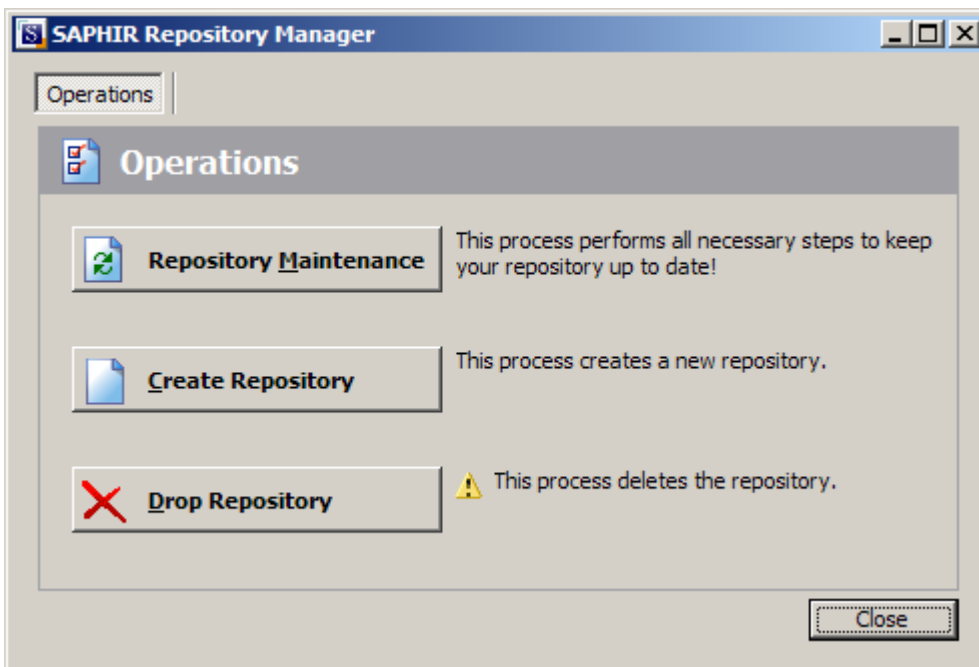
For an Oracle EBS system, see 'Connecting to Oracle Enterprise Business Suite' below.

Once the Repository Wizard Steps have been completed, Saphir performs a check on the database to see if the Repository tables exist. If not (or if the Repository was created with an earlier Saphir version), Saphir displays a form asking you to confirm that the Repository Maintenance function should be started. See the next section for details of the repository maintenance process. Note: for a SQLite-based Saphir repository, the Repository tables are created as part of the database creation process.

Note that you can return to the Repository Wizard forms at any time to amend the entered settings by clicking the  button on the Repository manager toolbar.

## 2.7 Repository Maintenance

The Repository Maintenance form deals with the Creation, Deletion and Modification of the tables and other RDBMS objects in a Saphir repository. The process will appear automatically when configuring a new repository, or when upgrading from an older version of Saphir. In addition, the Repository Maintenance form can be accessed by clicking the  button on the Repository manager toolbar.



Repository Maintenance

There are three buttons on the form:

- **Repository Maintenance:** Click this button to check the structure of the Saphir repository against the latest standard. This will typically be used when upgrading from an earlier Saphir release
- **Create Repository:** Click this button to create the tables, views and triggers that form the structure of the Saphir repository.
- **Drop Repository:** Click this button to drop all the tables, views and triggers in an existing Saphir repository. Please note that this will delete the entire contents of the repository!

In each case, Saphir executes a set of SQL scripts to perform the required database tasks.

## 2.8 Configuring Saphir to connect to an SAP system

One of the most important steps in installing Saphir for use with SAP is to decide which of two methods to use for extracting the metadata. The choices are:

- Extract via Direct Database Connection to SAP
- Extract via ABAP Function

Extracting via Direct Database Connection to SAP uses the appropriate RDBMS connectivity software to attach to the SAP Database to extract metadata via SQL calls from the SAP Data Dictionary Tables.

Extracting via ABAP uses an ABAP function, supplied with Saphir, to extract metadata into flat file format, via the SAP Application Layer. Saphir then takes the flat files and imports the metadata into the Saphir Repository.

The choice of Extraction Method influences the steps required to successfully implement Saphir. These are described in the following sections.

**IMPORTANT NOTE:** Extracting via Direct Database Connection to SAP will *NOT* extract the SAP Attribute definitions, as these are stored in a form not accessible to this connection method. If Attribute Definitions are required, use the ABAP Extraction method.

### 2.8.1 Connecting to SAP using the Direct Connection Method

If you have chosen to connect to SAP using the 'Direct' connection method, then it will be necessary to configure a connection to the SAP database where the required metadata is stored. As with the connection to the Saphir database, this is achieved using Microsoft ADO (ActiveX Data Objects).

### 2.8.2 Connecting to SAP using the ABAP Method

If you have chosen to connect to SAP using the ABAP connection method, then it will be necessary to register an ABAP Function on the SAP Application Server. There are two ABAP functions supplied as source code with the Saphir software. These will normally be in the Program Files\Silwood\Saphir 5\ABAPFiles folder as follows:

- For a SAP system: z\_saphir\_rfcs\_ddexpv5\_guidownload.txt.
- For a SAP BW system: z\_saphir\_rfcs\_bwexpv5\_guidownload.txt.

Full details of how to install and activate the ABAP function are to be found in Appendix A of this manual.

The steps described in the Appendices can be summarized as:

- Create a Package (sometimes known as a Development Class in earlier SAP releases)
- Create a Function Group
- Register the ABAP
- Test the ABAP

Note: Saphir is supplied with 2 additional ABAP functions (one each for SAP and SAP BW) that can be used to run the 'main' ABAP functions from within the SAP environment. These may be used to run the

extraction process and create text files if it is not possible to access the SAP system via RFC calls. See Appendix E for details of installing and using these functions.

### 2.8.3 Required Access Rights for the SAP User used by Saphir

If the extraction of metadata is to be run via ABAP and from within the Saphir environment then an SAP Username and Password will be required to enable Saphir to invoke the ABAP function.

This user must be of User Type 'Dialog' or 'Service' and have a security profile which allows RFC access to the Saphir Function Group created above. The SAP-delivered Profile S\_ABAP\_ALL would be sufficient.

Once the ABAP has been installed, you may proceed to the next stage. Turn to 'Entering the Extraction Method choice into Saphir' below.

### 2.8.4 Entering the extraction method choice into Saphir

Once you have decided the extraction method and configured the appropriate connection to the SAP system, navigate to Step 4 of the Repository Wizard. This is where you record the Database type being used by SAP, and the connection method.

The screenshot shows a dialog box titled "Step 4: Database for SAP". The main instruction is "Please define the type of the source database." There are two sections: "Database Type" and "Connection Type". In the "Database Type" section, there are three radio buttons: "Oracle" (selected), "DB2", and "MS SQL". In the "Connection Type" section, there are two radio buttons: "Direct DB Connect" and "SAP RFC Connect" (selected). At the bottom, there is a checked checkbox for "PC-Like Byte Order / Application Server on PC and not on a Workstation".

Repository Wizard – Step 4 – specifying the SAP database type and connection method

1. Select the radio button which corresponds to the database type on which the SAP system has been installed
2. For Microsoft SQL Server and DB2, a 'Command Timeout' property may be specified. This indicates the length of time to allow for a given query to complete. The default is 210 seconds. It will not normally be necessary to enter a value for this field unless 'timeout' problems are encountered.
3. Select the radio button for the Connection Type
4. If the SAP Application Server is installed on a PC rather than a non-windows workstation, check the 'PC-Like Byte Order...' check box.

Click the 'Next' button.

If you have chosen the 'Direct DB Connect' method for extracting metadata from SAP, the next form will be Step 5 – SAP Database Connection (see 'Entering the SAP Database Connection settings' below).

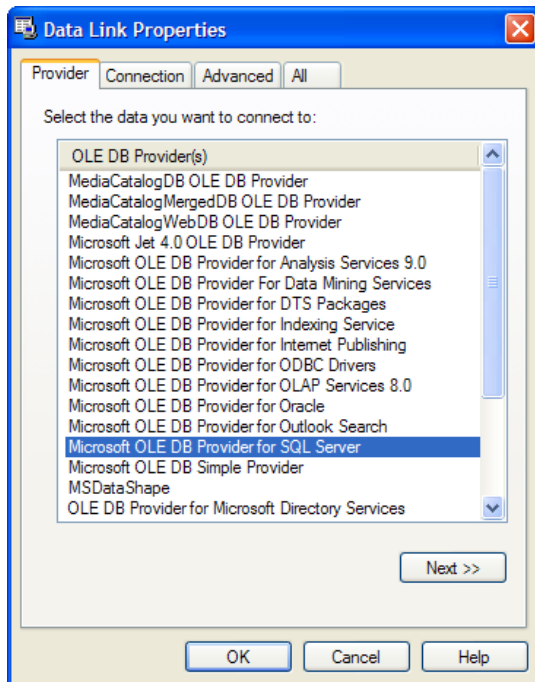
If you have chosen 'SAP RFC Connect', the next form will be Step 5 – SAP RFC Connection (see 'Entering the SAP RFC Connection settings' below).

## 2.8.5 Entering the SAP database connection settings

Repository Wizard – Step 5 – SAP Database Connection

Step 5 of the Repository Wizard records the information required for Saphir to connect directly to the SAP database.

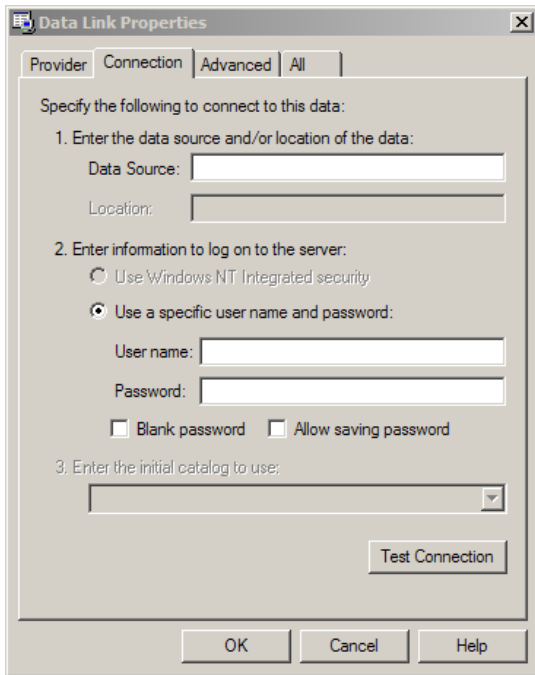
1. The Database Alias dropdown list shows all the existing ADO connections previously configured. If this is a new Saphir installation, it will be necessary to create a new connection to the SAP database. Click on the 'New' button to the right of the drop down to begin this process.
2. The 'Edit Connection Definition' form is now displayed. Enter an appropriate name for the connection in the 'Connection Definition name' field and click the 'OK' button. The ADO Data Link Properties form is then displayed.



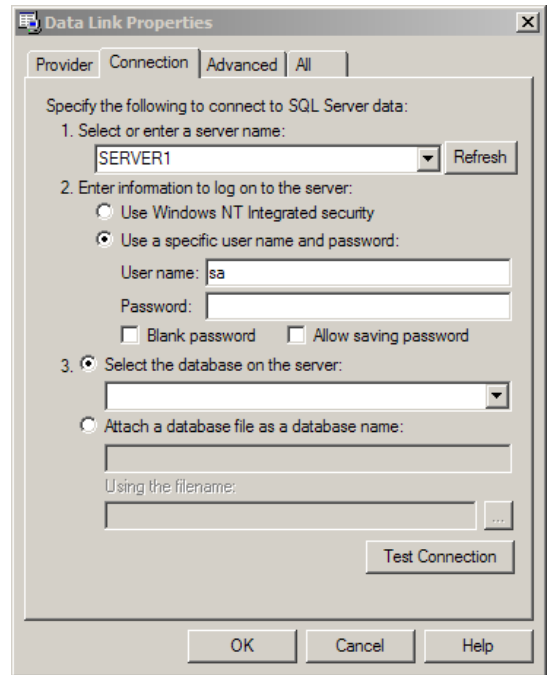
ADO Data Link Properties Form

3. If the database connection being created is Oracle, select 'Oracle Provider for OLE DB' from the list of Providers. If the database connection being created is SQL Server, select 'Microsoft OLE DB Provider for SQL Server'. For DB2 select the 'IBM OLEDB Provider for DB2'. Now click the 'Next' button.

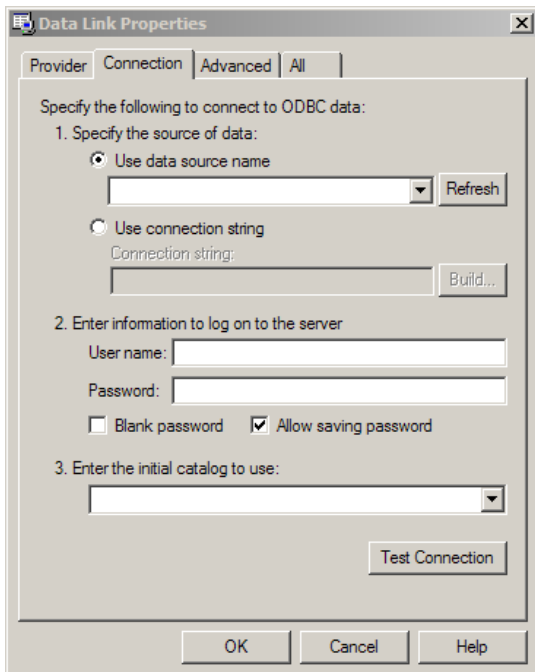
- The Data Link Properties Connection tab is now displayed. The available fields on this tab will be dependent on the chosen Provider. The layouts for Oracle, SQL Server and DB2 are shown below.



Data Link Properties 'Connection' tab for Oracle...



...for SQL Server...



...and for DB2

- For an Oracle database, in the 'Data Source' field, enter the connection string for connecting to the Oracle database. This will normally be the string used in the TNSNAMES file for connection to the database. For a SQL Server database, in the 'Select or enter a server name' field, enter or

select the name of the Server where the database is located. If the database is DB2, enter the database name in the 'Data Source'.

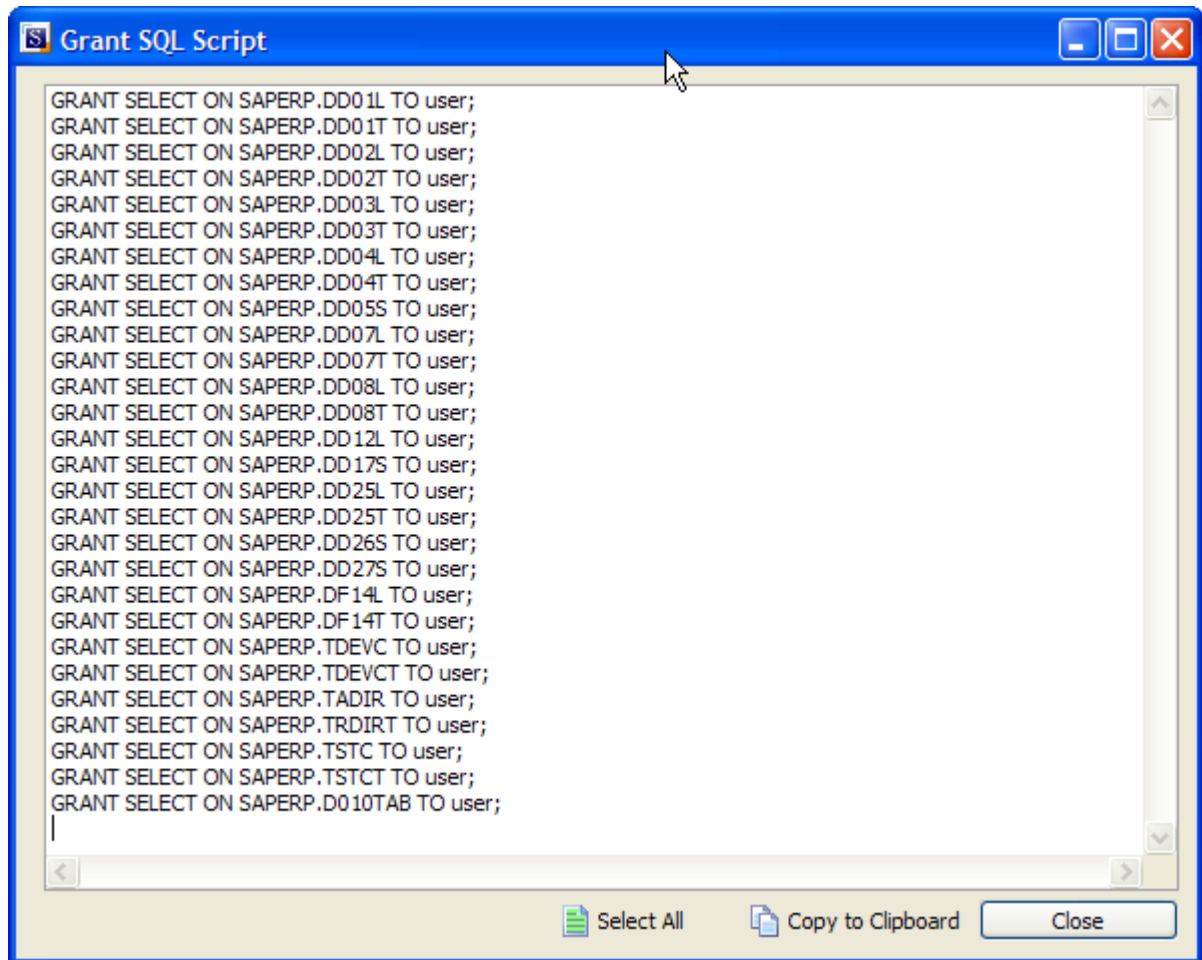
6. Enter a User name and Password. The User Name is a valid database User-id with 'select' access rights to the SAP database. Alternatively, leave these fields blank and enter the User Name and Password in the Repository Wizard form. (Note: this is a database Username and Password – not a SAP Application User Name and Password).
7. If a User name and Password were entered in the previous step, test the connection to the database by clicking the 'Test Connection' button.. If the connection is successful, click 'OK' to return to the Repository Wizard.
8. If not previously entered in the ADO configuration, enter a username and password for the database
9. If necessary, enter the SAP Table Owner (this is normally 'SAPR3').

Click the 'Test Connection' button to have Saphir check that the database can be accessed using the supplied information.

The 'Grant SQL' button will generate the appropriate SQL 'Grant' statements for each of the SAP tables that Saphir need to access in the SAP database. You can use this feature to help your SAP DBA set up a database user for use in Saphir.

Click the 'Save' button to save all the settings entered during the Repository Wizard session.

Now turn to ‘Opening a Saphir Repository’ below.



The Grant SQL form showing the GRANT statements for each SAP table accessed by Saphir

## 2.9 Entering the SAP RFC connection settings

**Step 5: SAP RFC Connection**

Please define and test the RFC Connection of the source system.

**RFC Connection**

Application Server:       SAP Client No.:

System Number:       SAP Username:

SAP RFC.ini DEST:  (optional)      SAP Password:

SAP GUI Version:       SAP Language:

RFC Function Name:

RFC Extract Path:  ...

Repository Wizard – Step 5 – SAP RFC Connection

Step 5 of the Repository Wizard records the information required for Saphir to connect to the SAP system using SAP Remote Function Calls (RFCs). If the ABAP function is being run external to the Saphir environment (see Appendix E: Using the ABAP ‘ALL’ Functions) then there is no need to enter this information.

1. Enter the Application Server name of the desired SAP system, the SAP Client Number, the System Number and the Username and Password.
2. Enter the optional SAP RFC.ini Destination string if the SAP RFC.ini file is being used for connection to SAP systems (see Appendix C for more details of the SAP RFC.ini usage).
3. Use the SAP GUI Version drop down to choose the SAP GUI version of 4.x, 6.x or 7.x.
4. Select the desired SAP extraction language. If you do not see the language you require, you can modify the range of available language codes by modifying the SaphirSettings.xls file. (see Appendix B for details).
5. Specify the name of the ABAP Function to be invoked by the RFC calls (see Appendix A for details of the ABAP function )
6. Enter a path for the location of the RFC text files. This is only required if the extraction is to be performed in two phases, as described in ‘Extracting Metadata from SAP’ in Chapter 3 of this manual.

Click the ‘Test Connection’ button to have Saphir check that the SAP system can be accessed using the supplied information.

Click the ‘Save’ button to save all the settings entered during the Repository Wizard session. Now turn to ‘Opening a Saphir Repository’ below.

## 2.10 Connecting to PeopleSoft Enterprise

If you have chosen to connect to a PeopleSoft system, then it will be necessary to configure a connection to the PeopleSoft database where the required metadata is stored.

As with the connection to the Saphir database, this is achieved using Microsoft ADO (ActiveX Data Objects).

### 2.10.1 Choosing the PeopleSoft database type

Now navigate to Step 4 of the Repository Wizard. This is where you record the Database type being used by PeopleSoft.

**Step 4: Database for PeopleSoft Enterprise**

Please define the type of the source database.

**Database Type**

Oracle  DB2  MS SQL

**Database Connection**

Command Timeout:  [sec]

Repository Wizard – Step 4 – specifying the PeopleSoft database type

Select the radio button which corresponds to the database type on which the PeopleSoft system has been installed.

For Microsoft SQL Server and DB2, a ‘Command Timeout’ property may be specified. This indicates the length of time to allow for a given query to complete. The default is 210 seconds. It will not normally be necessary to enter a value for this field unless ‘timeout’ problems are encountered.

Click the ‘Next’ button.

## 2.10.2 Entering the PeopleSoft database connection settings

Step 5 of the Repository Wizard records the information required for Saphir to connect to the PeopleSoft database.

**Step 5: PeopleSoft Database Connection**

Please define and test the database connection of the source system.

**Database Connection**

Database Alias:

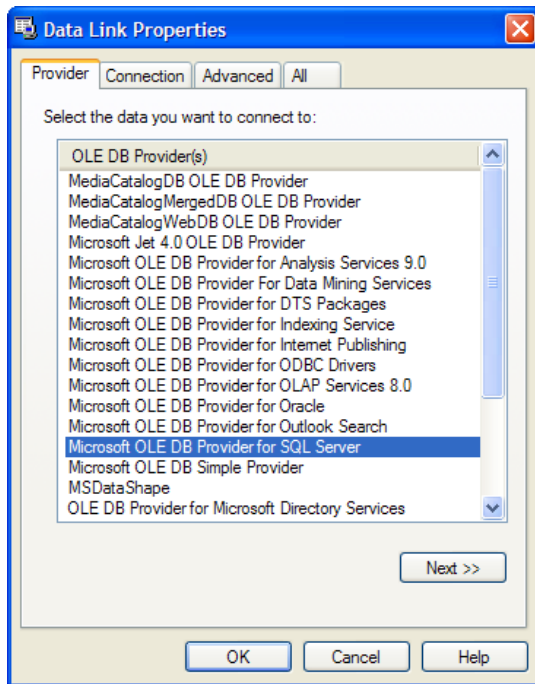
Database Username:

Database Password:

Table Owner:

Repository Wizard – Step 5 – PeopleSoft Database Connection

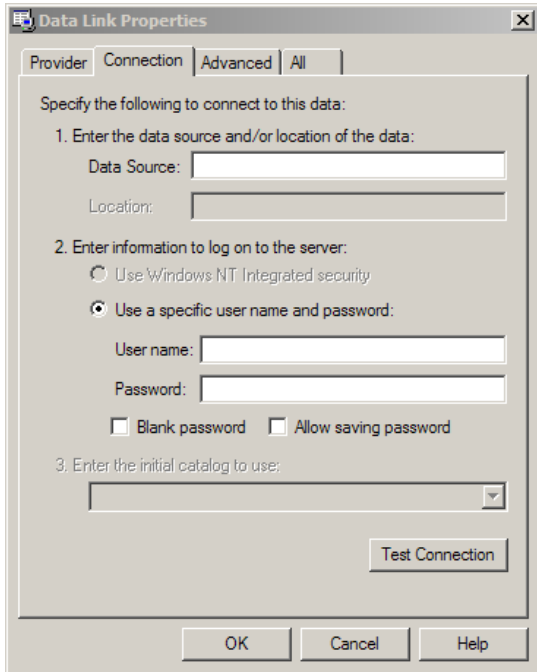
1. The Database Alias dropdown list shows all the existing ADO connections previously configured. If this is a new Saphir installation, it will be necessary to create a new connection to the PeopleSoft database. Click on the 'New' button to the right of the drop down to begin this process.
2. The 'Edit Connection Definition' form is now displayed. Enter an appropriate name for the connection in the 'Connection Definition name' field and click the 'OK' button. The ADO Data Link Properties form is then displayed.



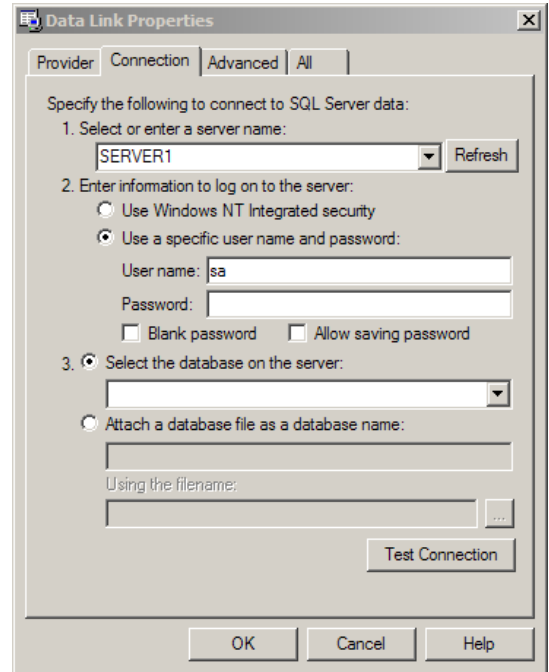
ADO Data Link Properties Form

3. If the database connection being created is Oracle, select 'Oracle Provider for OLE DB' from the list of Providers. If the database connection being created is SQL Server, select 'Microsoft OLE DB Provider for SQL Server'. For DB2 select the 'IBM OLEDB Provider for DB2'. Now click the 'Next' button.

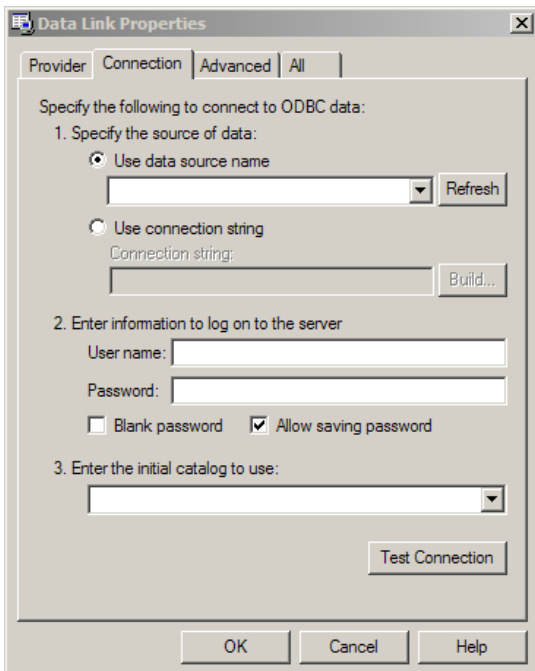
- The Data Link Properties Connection tab is now displayed. The available fields on this tab will be dependent on the chosen Provider. The layouts for Oracle, SQL Server and DB2 are shown below.



Data Link Properties 'Connection' tab for Oracle...



...for SQL Server...



...and for DB2

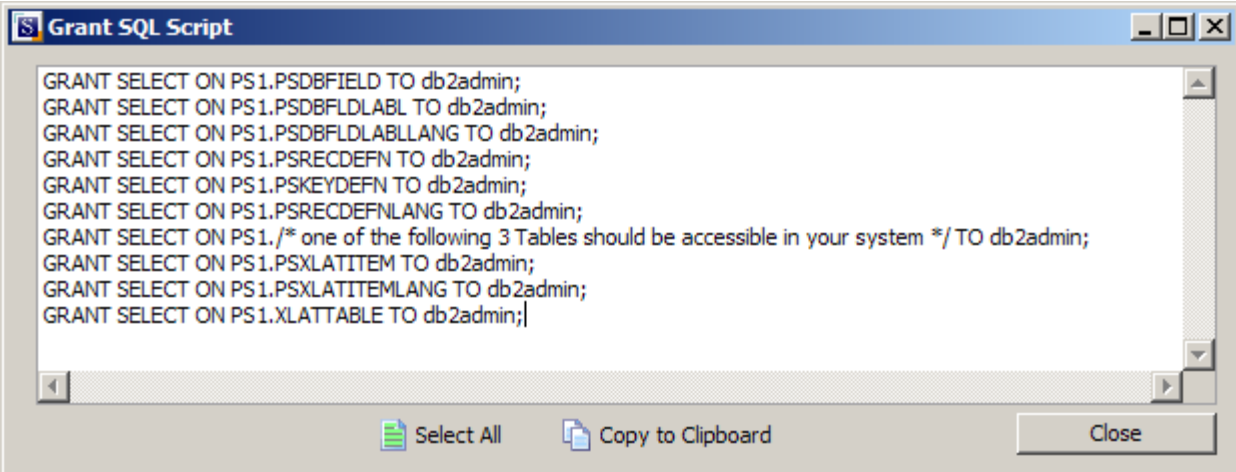
5. For an Oracle database, in the 'Data Source' field, enter the connection string for connecting to the Oracle database. This will normally be the string used in the TNSNAMES file for connection to the database. For a SQL Server database, in the 'Select or enter a server name' field, enter or select the name of the Server where the database is located. If the database is DB2, enter the database name in the 'Data Source'.
6. Enter a User name and Password. The User Name is a valid database User-id with 'select' access rights to the PeopleSoft database. Alternatively, leave these fields blank and enter the User Name and Password in the Repository Wizard form. (Note: this is a database Username and Password – not a PeopleSoft Application User Name and Password).
7. If a User name and Password were entered in the previous step, test the connection to the database by clicking the 'Test Connection' button.. If the connection is successful, click 'OK' to return to the Repository Wizard. (Note: this is a database Username and Password – not a PeopleSoft Application Username and Password).
8. If not previously entered in the ADO configuration, enter a username and password for the database
9. If necessary, enter the PeopleSoft Table Owner

Click the 'Test Connection' button to have Saphir check that the database can be accessed using the supplied information.

The 'Grant SQL' button will generate the appropriate SQL 'Grant' statements for each of the PeopleSoft tables that Saphir need to access in the PeopleSoft database. You can use this feature to help your PeopleSoft DBA set up a database User for you.

Click the 'Save' button to save all the settings entered during the Repository Wizard session.

Now turn to 'Opening a Saphir Repository' below.



```

Grant SQL Script
GRANT SELECT ON PS1.PSDBFIELD TO db2admin;
GRANT SELECT ON PS1.PSDBFLDLABL TO db2admin;
GRANT SELECT ON PS1.PSDBFLDLABLLANG TO db2admin;
GRANT SELECT ON PS1.PSRECDEFN TO db2admin;
GRANT SELECT ON PS1.PSKEYDEFN TO db2admin;
GRANT SELECT ON PS1.PSRECDEFNLANG TO db2admin;
GRANT SELECT ON PS1./* one of the following 3 Tables should be accessible in your system */ TO db2admin;
GRANT SELECT ON PS1.PSXLATITEM TO db2admin;
GRANT SELECT ON PS1.PSXLATITMLANG TO db2admin;
GRANT SELECT ON PS1.XLATTABLE TO db2admin;
  
```

Select All Copy to Clipboard Close

The Grant SQL form showing the GRANT statements for each PeopleSoft table accessed by Saphir

## 2.11 Connecting to J.D.Edwards EnterpriseOne

If you have chosen to connect to a J.D.Edwards EnterpriseOne system, then it will be necessary to configure a connection to the EnterpriseOne database where the required metadata is stored. Saphir extracts the metadata from a small set of tables in this database and a group of XML files that are generated by a process on the EnterpriseOne application. You will need to generate the XML files before attempting the extraction of metadata from the EnterpriseOne system.

As with the connection to the Saphir database, the connection to the EnterpriseOne database is achieved using Microsoft ADO (ActiveX Data Objects). The tables that Saphir accesses are EnterpriseOne Data Dictionary tables (in particular, F9202, F9210 and F00165).

### 2.11.1 To produce the XML Files from the EnterpriseOne system

1. From the J.D.Edwards EnterpriseOne Explorer, select Foundation and then Report Writer.
2. From the Menu, select Batch Versions.
3. Key R91400A into the Batch Application field and press Find. There is only one version, select it by double clicking.

The XML files produced should be:

- SysCodes.xml
- Tables.xml
- UseCodes.xml
- Views.xml

## 2.11.2 Choosing the EnterpriseOne database type

Now navigate to Step 4 of the Repository Wizard. This is where you record the Database type being used by the EnterpriseOne system.

Step 4: Database for JD Edwards EnterpriseOne	
Please define the type of the source database.	
Database Type	
<input type="radio"/> Oracle	<input type="radio"/> DB2
<input checked="" type="radio"/> MS SQL	
Database Connection	
Command Timeout:	<input type="text"/> [sec]

Repository Wizard – Step 4 – specifying the EnterpriseOne database type

Select the radio button which corresponds to the database type on which the EnterpriseOne system has been installed

For Microsoft SQL Server and DB2, a ‘Command Timeout’ property may be specified. This indicates the length of time to allow for a given query to complete. The default is 210 seconds. It will not normally be necessary to enter a value for this field unless ‘timeout’ problems are encountered.

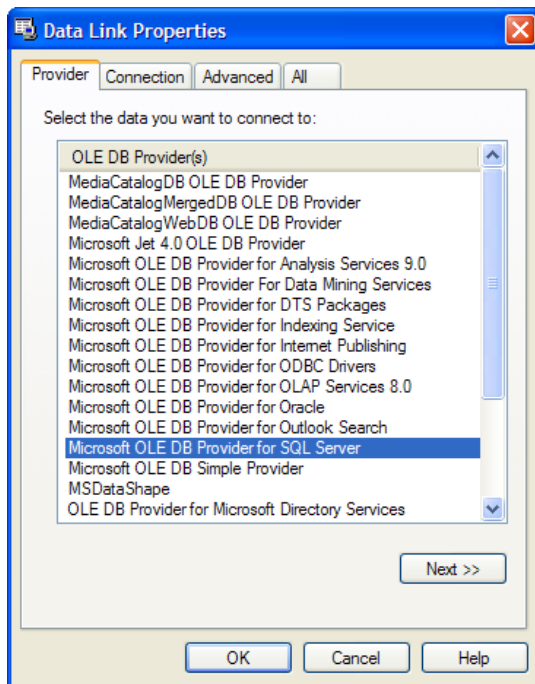
Click the ‘Next’ button.

### 2.11.3 Entering the EnterpriseOne database connection settings

Step 5 of the Repository Wizard records the information required for Saphir to connect to the EnterpriseOne database.

Repository Wizard – Step 5 – EnterpriseOne Database Connection

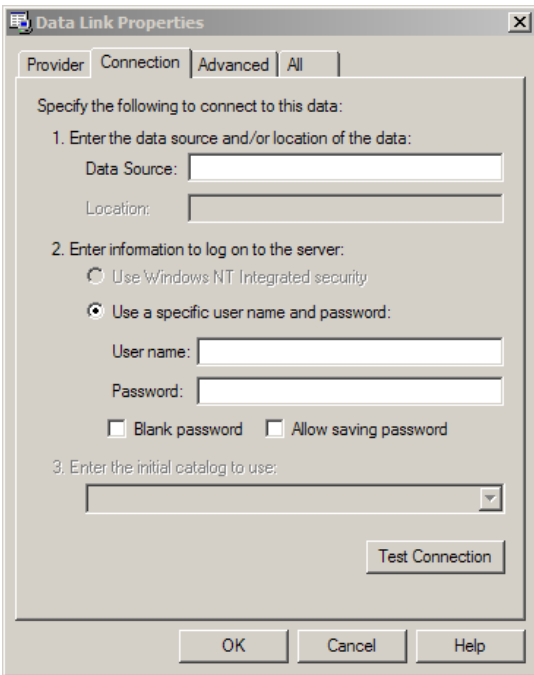
1. The Database Alias dropdown list shows all the existing ADO connections previously configured. If this is a new Saphir installation, it will be necessary to create a new connection to the EnterpriseOne database. Click on the 'New' button to the right of the drop down to begin this process.
2. The 'Edit Connection Definition' form is now displayed. Enter an appropriate name for the connection in the 'Connection Definition name' field and click the 'OK' button. The ADO Data Link Properties form is then displayed.



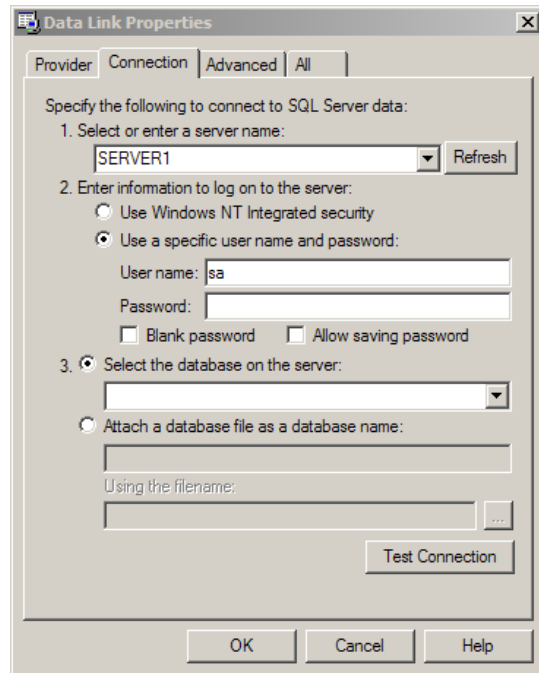
ADO Data Link Properties Form

3. If the database connection being created is Oracle, select 'Oracle Provider for OLE DB' from the list of Providers. If the database connection being created is SQL Server, select 'Microsoft OLE DB Provider for SQL Server'. For DB2 select the 'IBM OLEDB Provider for DB2'. Now click the 'Next' button.

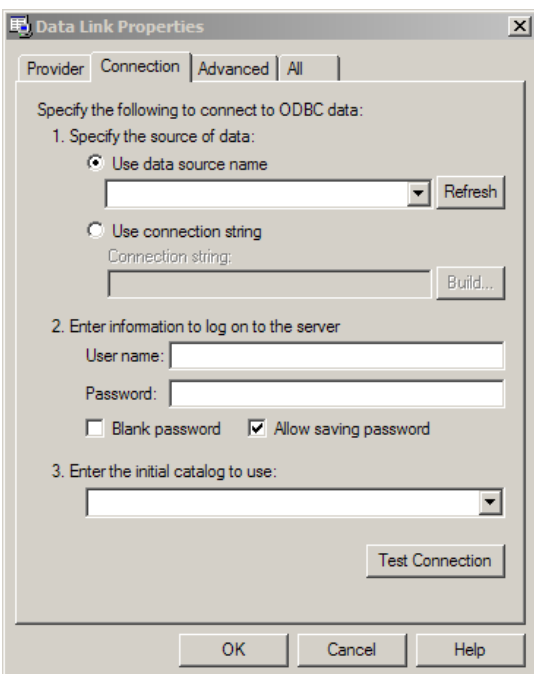
- The Data Link Properties Connection tab is now displayed. The available fields on this tab will be dependent on the chosen Provider. The layouts for Oracle, SQL Server and DB2 are shown below.



Data Link Properties 'Connection' tab for Oracle...



...for SQL Server...



...and for DB2

5. For an Oracle database, in the 'Data Source' field, enter the connection string for connecting to the Oracle database. This will normally be the string used in the TNSNAMES file for connection to the database. For a SQL Server database, in the 'Select or enter a server name' field, enter or select the name of the Server where the database is located. If the database is DB2, enter the database name in the 'Data Source'.
6. Enter a User name and Password. The User Name is a valid database User-id with 'select' access rights to the EnterpriseOne database. Alternatively, leave these fields blank and enter the User Name and Password in the Repository Wizard form. (Note: this is a database Username and Password – not an EnterpriseOne Application User Name and Password).
7. If a User name and Password were entered in the previous step, test the connection to the database by clicking the 'Test Connection' button.. If the connection is successful, click 'OK' to return to the Repository Wizard.
8. If not previously entered in the ADO configuration, enter a username and password for the database
9. If necessary, enter the EnterpriseOne Table Owner.
10. Enter the directory where the 4 XML files produced from the EnterpriseOne batch process described in Connecting to J.D.Edwards EnterpriseOne' above are located.

Click the 'Test Connection' button to have Saphir check that the database can be accessed using the supplied information.

Click the 'Save' button to save all the settings entered during the Repository Wizard session.

Now turn to 'Opening a Saphir Repository' below.

## 2.12 Connecting to Siebel

If you have chosen to connect to a Siebel system, then it will be necessary to configure a connection to the Siebel database where the required metadata is stored.

As with the connection to the Saphir database, this is achieved using Microsoft ADO (ActiveX Data Objects).

### 2.12.1 Choosing the Siebel database type

Now navigate to Step 4 of the Repository Wizard. This is where you record the Database type being used by Siebel.

**Step 4: Database for Siebel**

Please define the type of the source database.

**Database Type**

Oracle  DB2  MS SQL

**Database Connection**

Command Timeout:  [sec]

Repository Wizard – Step 4 – specifying the Siebel database type

Select the radio button which corresponds to the database type on which the Siebel system has been installed.

For Microsoft SQL Server and DB2, a 'Command Timeout' property may be specified. This indicates the length of time to allow for a given query to complete. The default is 210 seconds. It will not normally be necessary to enter a value for this field unless 'timeout' problems are encountered.

Click the 'Next' button.

## 2.12.2 Entering the Siebel database connection settings

Step 5 of the Repository Wizard records the information required for Saphir to connect to the Siebel database.

**Step 5: Siebel Database Connection**

Please define and test the database connection of the source system.

**Database Connection**

Connection Name: SIEBELDB

Database Username: sa

Database Password: \*\*\*\*

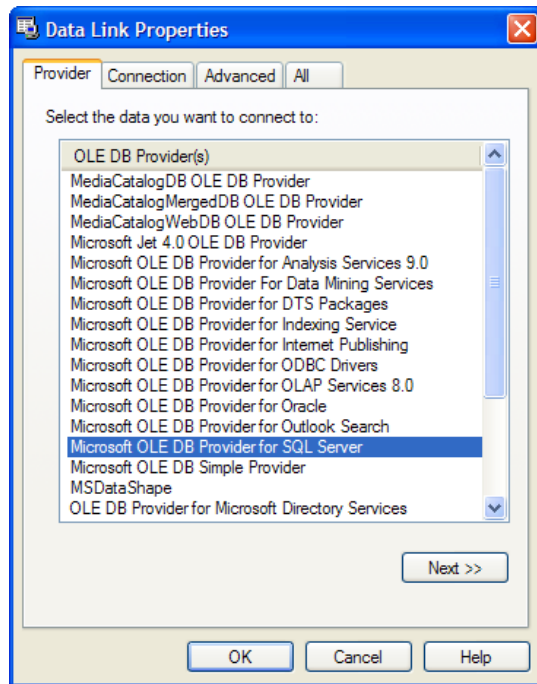
Table Owner: S1

**Siebel Repository Settings**

Repository Name: 1-4-1 | Siebel Repository

Repository Wizard – Step 5 – Siebel Database Connection

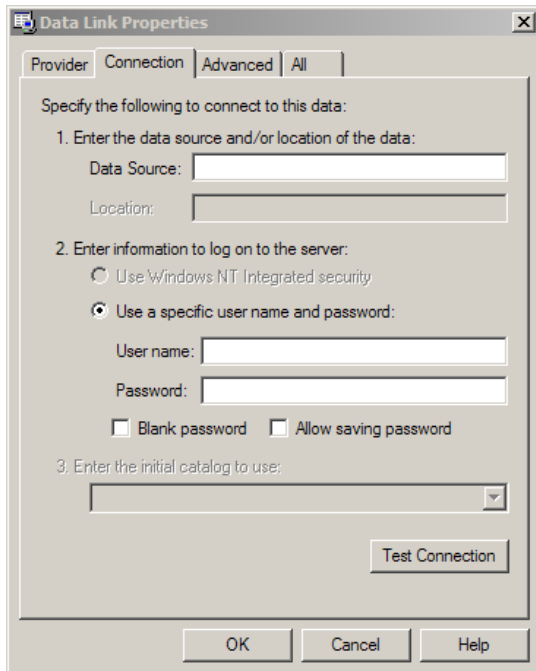
1. The Database Alias dropdown list shows all the existing ADO connections previously configured. If this is a new Saphir installation, it will be necessary to create a new connection to the Siebel database. Click on the 'New' button to the right of the drop down to begin this process.
2. The 'Edit Connection Definition' form is now displayed. Enter an appropriate name for the connection in the 'Connection Definition name' field and click the 'OK' button. The ADO Data Link Properties form is then displayed.



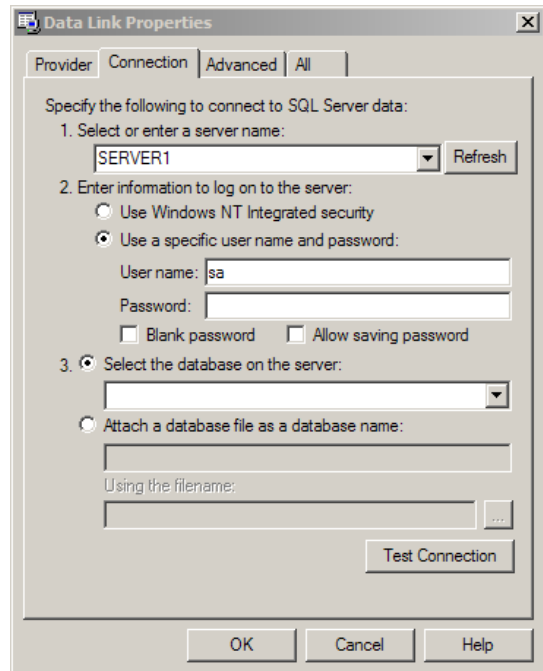
ADO Data Link Properties Form

3. If the database connection being created is Oracle, select 'Oracle Provider for OLE DB' from the list of Providers. If the database connection being created is SQL Server, select 'Microsoft OLE DB Provider for SQL Server'. For DB2 select the 'IBM OLEDB Provider for DB2'. Now click the 'Next' button.

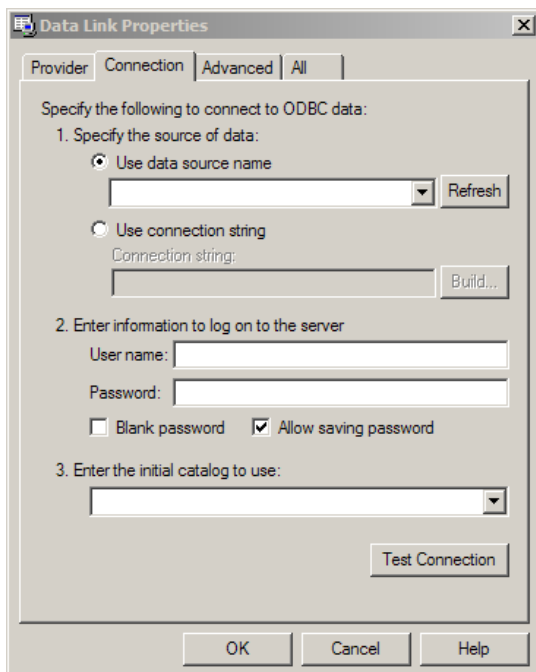
- The Data Link Properties Connection tab is now displayed. The available fields on this tab will be dependent on the chosen Provider. The layouts for Oracle, SQL Server and DB2 are shown below.



Data Link Properties 'Connection' tab for Oracle...



...for SQL Server...



...and for DB2

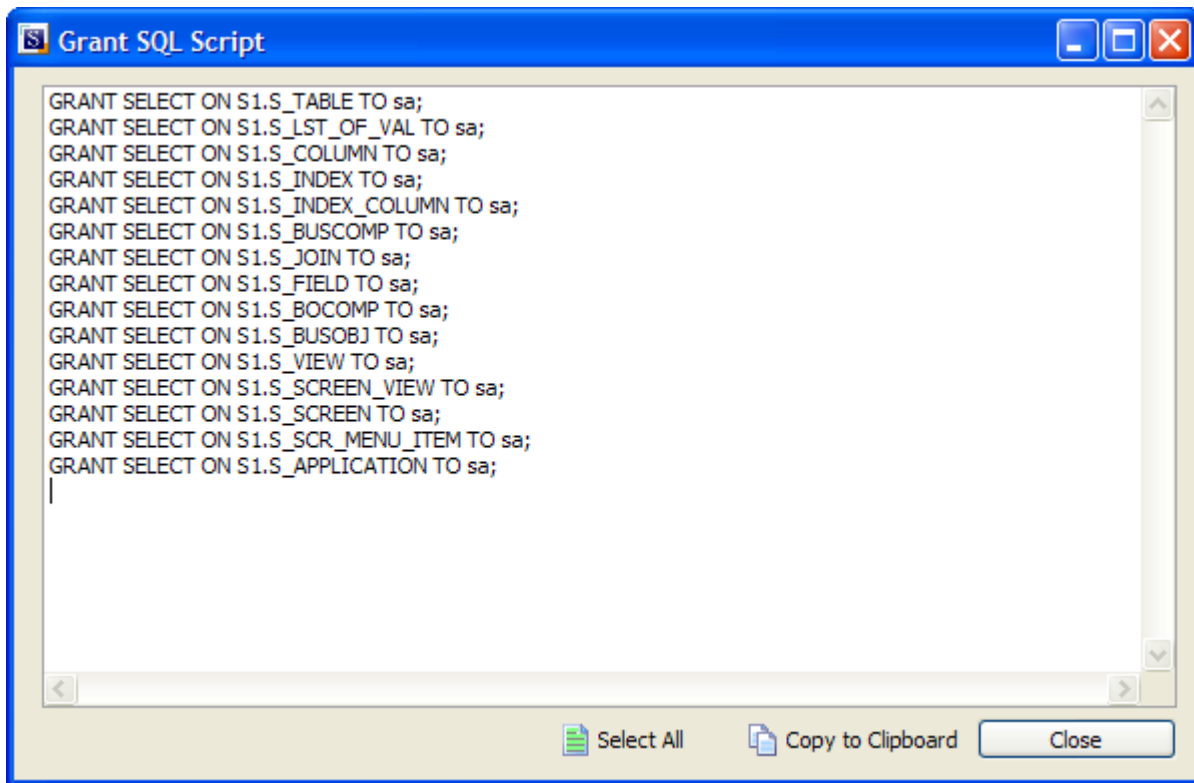
5. For an Oracle database, in the 'Data Source' field, enter the connection string for connecting to the Oracle database. This will normally be the string used in the TNSNAMES file for connection to the database. For a SQL Server database, in the 'Select or enter a server name' field, enter or select the name of the Server where the database is located. If the database is DB2, enter the database name in the 'Data Source'.
6. Enter a User name and Password. The User Name is a valid database User-id with 'select' access rights to the Siebel database. Alternatively, leave these fields blank and enter the User Name and Password in the Repository Wizard form. (Note: this is a database Username and Password – not a Siebel Application User Name and Password).
7. If a User name and Password were entered in the previous step, test the connection to the database by clicking the 'Test Connection' button.. If the connection is successful, click 'OK' to return to the Repository Wizard. (Note: this is a database Username and Password – not a Siebel Application Username and Password).
8. If not previously entered in the ADO configuration, enter a username and password for the database
9. Using the Repository drop down list, select the Siebel Repository from which the extraction is to be performed. (Note: this is the *Siebel* Repository, not the Saphir Repository).
10. If necessary, enter the Siebel Table Owner

Click the 'Test Connection' button to have Saphir check that the database can be accessed using the supplied information.

The 'Grant SQL' button will generate the appropriate SQL 'Grant' statements for each of the Siebel tables that Saphir need to access in the Siebel database. You can use this feature to help your Siebel DBA set up a database User for you.

Click the 'Save' button to save all the settings entered during the Repository Wizard session.

Now turn to 'Opening a Saphir Repository' below.



The Grant SQL form showing the GRANT statements for each Siebel table accessed by Saphir

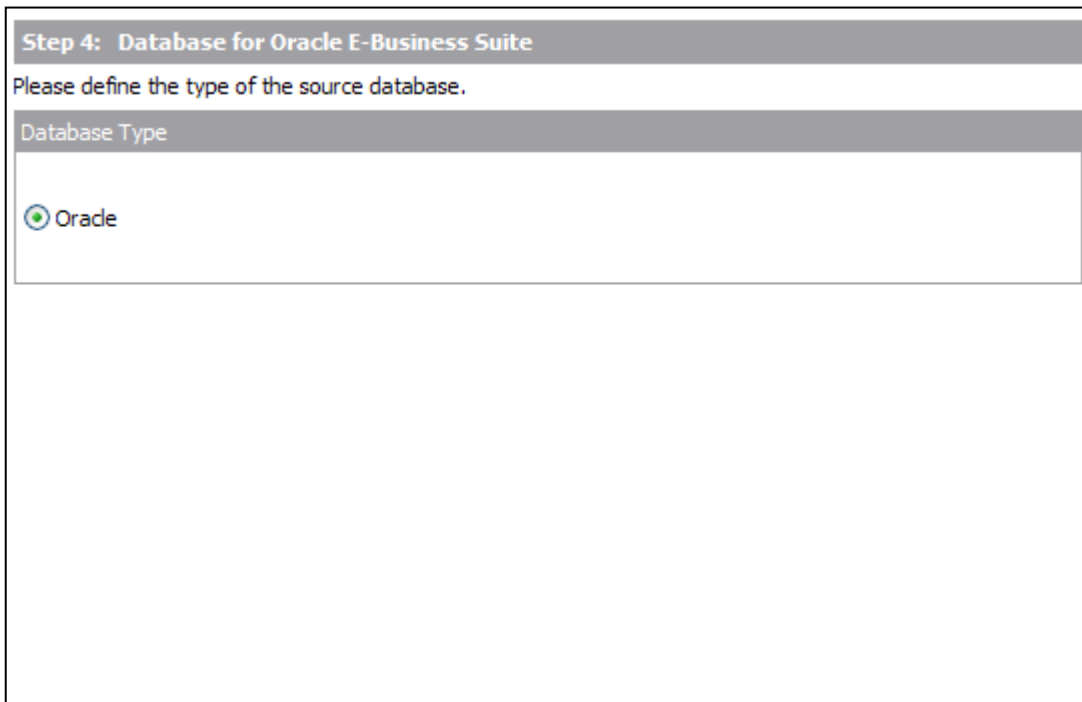
## 2.13 Connecting to Oracle Enterprise Business Suite

If you have chosen to connect to an Oracle EBS system, then it will be necessary to configure a connection to the Oracle EBS database where the required metadata is stored.

As with the connection to the Saphir database, this is achieved using Microsoft ADO (ActiveX Data Objects).

### 2.13.1 Choosing the EBS database type

Now navigate to Step 4 of the Repository Wizard. This is where you record the Database type being used by EBS.



Repository Wizard – Step 4 – specifying the EBS database type

In the case of EBS, there is only one possible source database type and this is pre-set as Oracle.

Click the 'Next' button.

## 2.13.2 Entering the EBS database connection settings

Step 5 of the Repository Wizard records the information required for Saphir to connect to the Oracle EBS database.

**Step 5: Oracle E-Business Suite Database Connection**

Please define and test the database connection of the source system.

**Database Connection**

Connection Name:

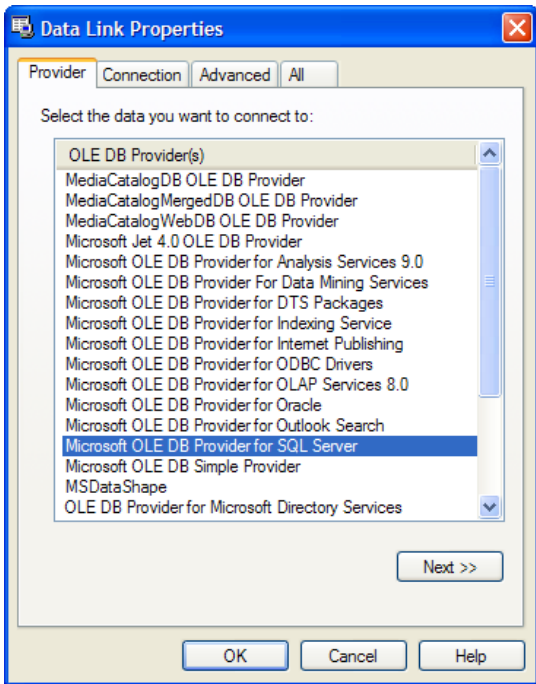
Database Username:

Database Password:

Table Owner:

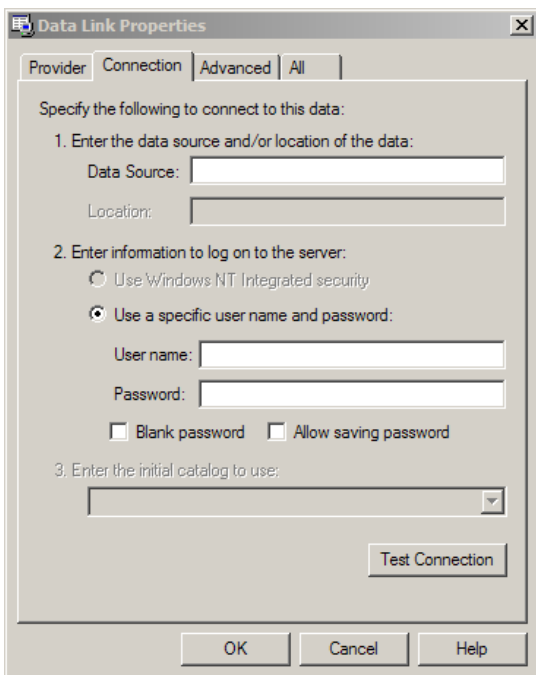
Repository Wizard – Step 5 – EBS Database Connection

10. The Database Alias dropdown list shows all the existing ADO connections previously configured. If this is a new Saphir installation, it will be necessary to create a new connection to the EBS database. Click on the 'New' button to the right of the drop down to begin this process.
11. The 'Edit Connection Definition' form is now displayed. Enter an appropriate name for the connection in the 'Connection Definition name' field and click the 'OK' button. The ADO Data Link Properties form is then displayed.



ADO Data Link Properties Form

12. Select 'Oracle Provider for OLE DB' from the list of Providers. Now click the 'Next' button.
13. The Data Link Properties Connection tab is now displayed.



Data Link Properties 'Connection' tab for Oracle

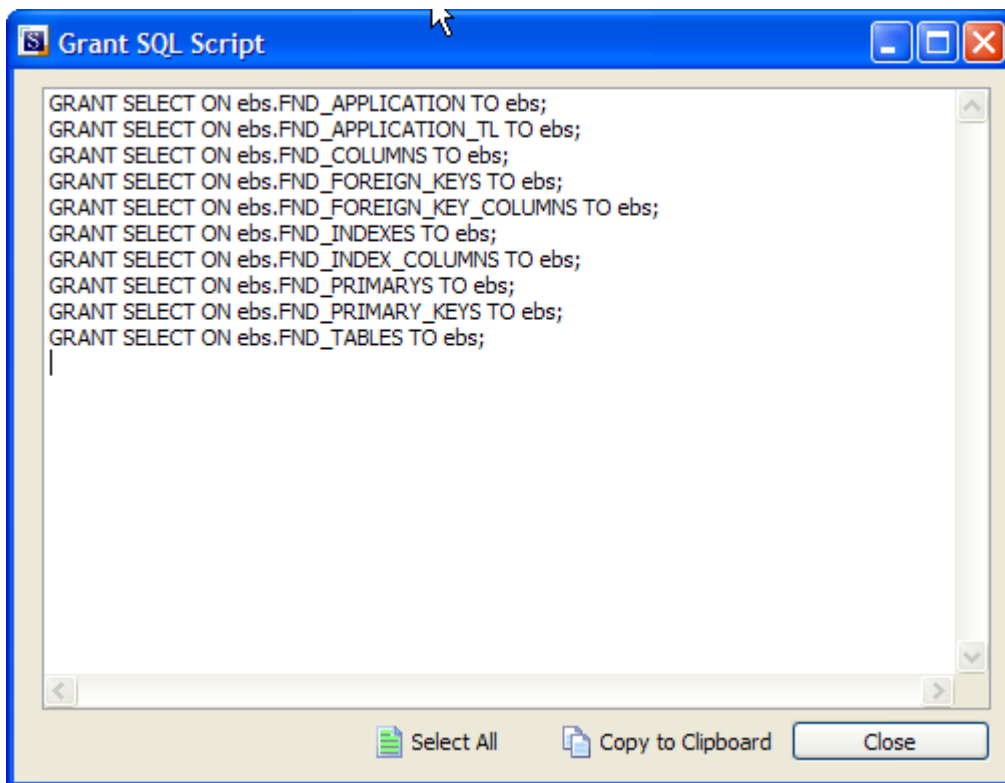
14. In the 'Data Source' field, enter the connection string for connecting to the Oracle database. This will normally be the string used in the TNSNAMES file for connection to the database.
15. Enter a User name and Password. The User Name is a valid database User-id with 'select' access rights to the Oracle EBS database. Alternatively, leave these fields blank and enter the User Name and Password in the Repository Wizard form. (Note: this is a database Username and Password – not an EBS Application User Name and Password).
16. If a User name and Password were entered in the previous step, test the connection to the database by clicking the 'Test Connection' button.. If the connection is successful, click 'OK' to return to the Repository Wizard. (Note: this is a database Username and Password – not an EBS Application Username and Password).
17. If not previously entered in the ADO configuration, enter a username and password for the database. If necessary, enter the EBS database Table Owner

Click the 'Test Connection' button to have Saphir check that the database can be accessed using the supplied information.

The 'Grant SQL' button will generate the appropriate SQL 'Grant' statements for each of the EBS tables that Saphir need to access in the EBS database. You can use this feature to help your EBS DBA set up a database User for you.

Click the 'Save' button to save all the settings entered during the Repository Wizard session.

Now turn to 'Opening a Saphir Repository' below.




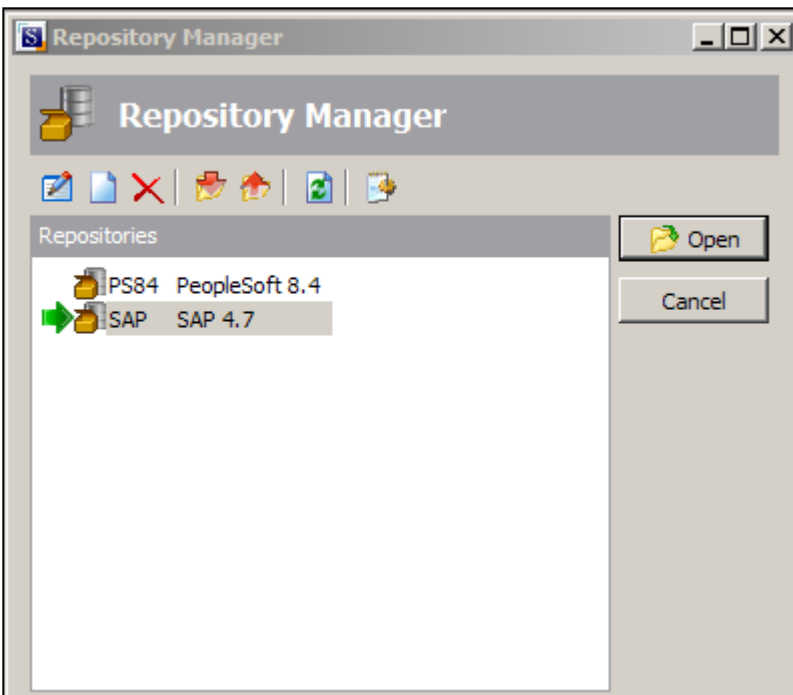
The Grant SQL form showing the GRANT statements for each EBS table accessed by Saphir

## 2.14 Opening a Saphir Repository

The Repository Manager is not only the means by which Saphir Repositories are created and maintained, it is also the mechanism for opening and switching between repositories.

### 2.14.1 To Open an existing Repository

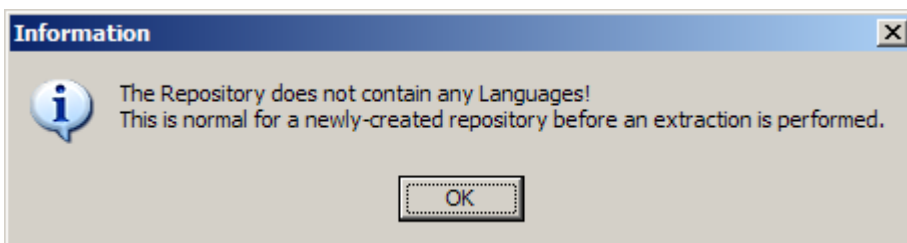
When Saphir is started, the Repository Manager is automatically displayed. If Saphir is already started and you wish to switch to a different repository, click the  button on the Saphir toolbar.



Selecting a Repository in the Repository Manager

To open a repository, select the required entry from the list and click the 'Open' button.

For an empty repository (where no extraction has yet been performed), the following message will be displayed.



This indicates that there are no data definitions stored in the repository as yet. The population of the repository occurs when you perform an extraction from your chosen Enterprise Application. This is described in the next chapter.

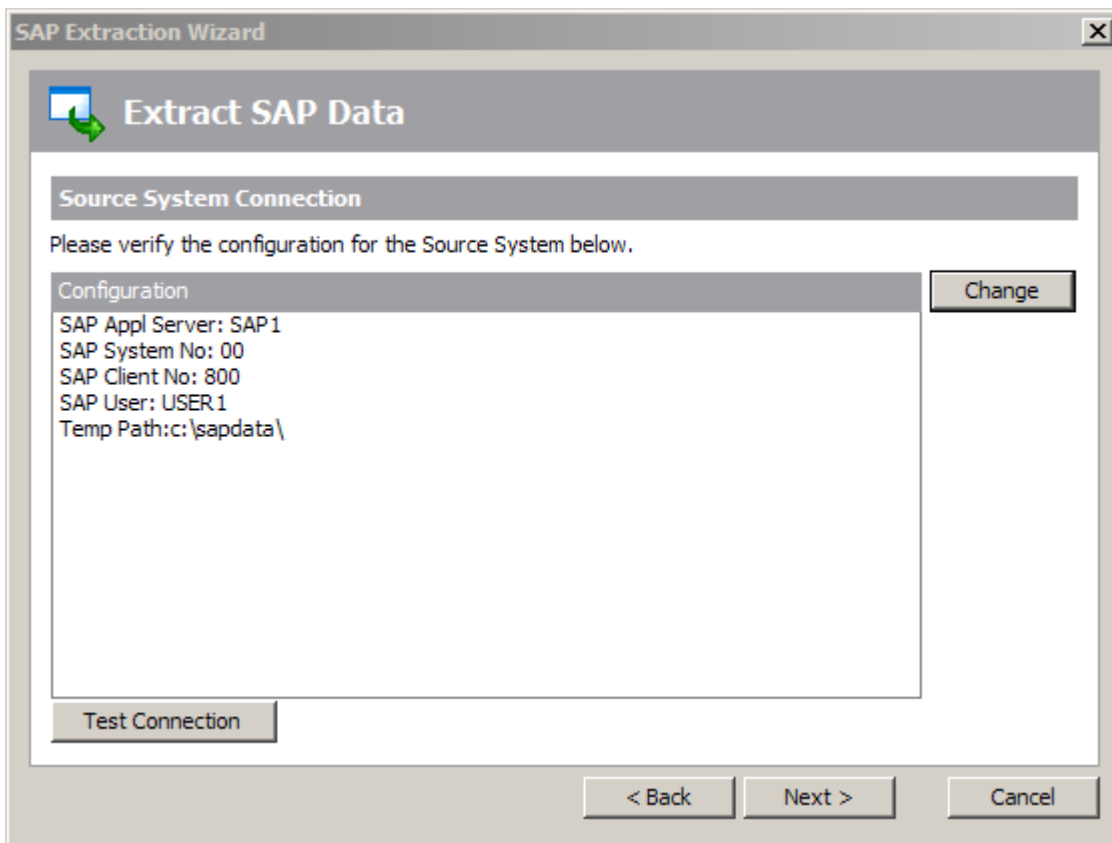
### 3 Extracting Metadata

*How to extract metadata from your chosen Enterprise Application*

**H**aving successfully installed Saphir, the next step is to extract metadata from the Enterprise Application. The following sections describe how to achieve this for SAP, PeopleSoft Enterprise, J.D.Edwards EnterpriseOne, Siebel and Oracle EBS.

#### 3.1 Extracting metadata from SAP or SAP BW

Start the extraction process by selecting the 'ERP Extract' option from the Saphir File menu. This will start the SAP Extraction Wizard. Click the 'Next' button on the first 'Extraction Wizard' form to display the 'Source System Connection' form.

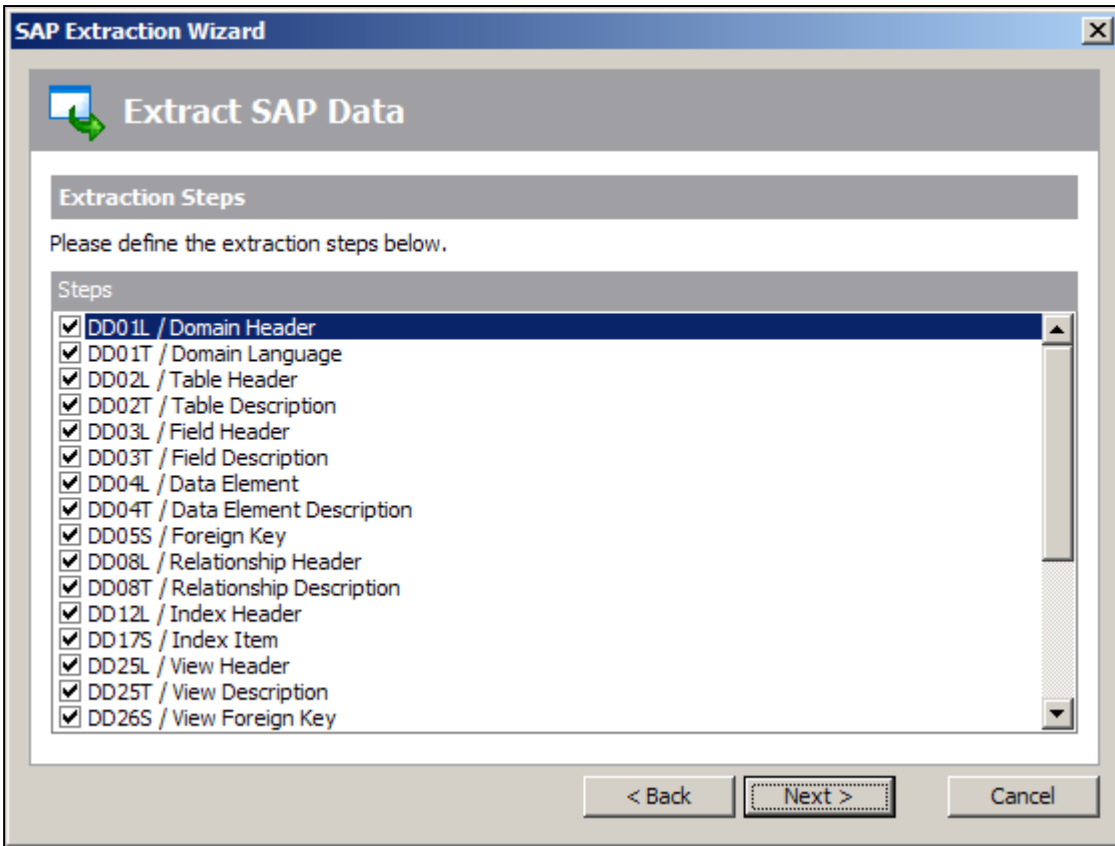


SAP Extraction Wizard – Source System Connection

This form shows a summary of the connection parameters, previously supplied when completing the Repository Wizard. The connection parameters will vary in format depending on the connection method chosen (Direct DB Connect or SAP RFC Connect). You can:

- Click the 'Change' button to further amend the connection parameters
- Click the 'Test Connection' button to verify the connection through to the Application.

Now click the 'Next' button to proceed to the next stage.

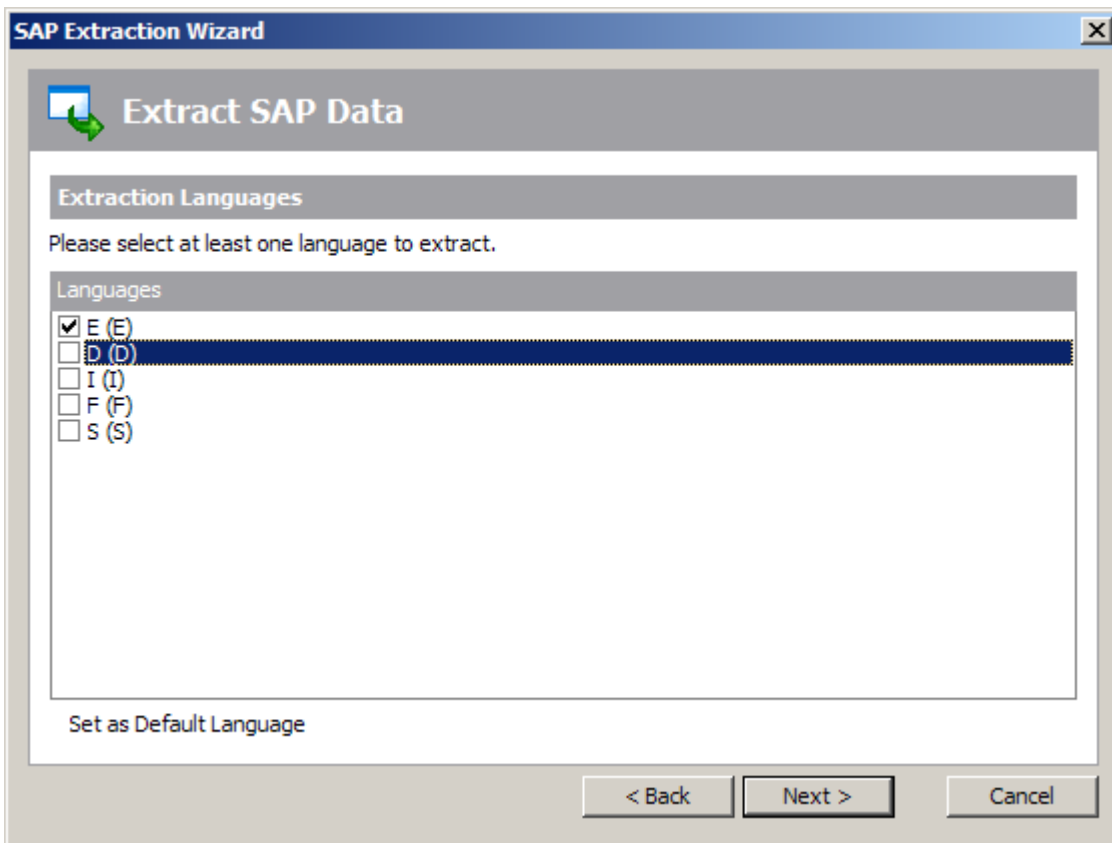


SAP Extraction Wizard – Extraction Steps

This form lists each of the stages involved in extracting metadata from an SAP system. The first 25 steps are checked ‘On’ and we recommend that you leave this selection and continue to the next stage. The 26<sup>th</sup> and last step is to ‘Generate Extended Relationships’. This describes the process by which Saphir infers additional relationships for the SAP metadata by examining likely Foreign Keys in the SAP table structures. This option is checked ‘Off’ by default and we recommend that you only select the option to be ‘On’ after fully understanding the Extended Relationship generation process which is described in the ‘Saphir User Guide.’

Note: The steps for a SAP BW system are different, but again, it is recommended that you leave the default selection of steps. There is no equivalent of the ‘Generate Extended relationships’ for BW.

Click the ‘Next’ button to proceed to the next stage’.

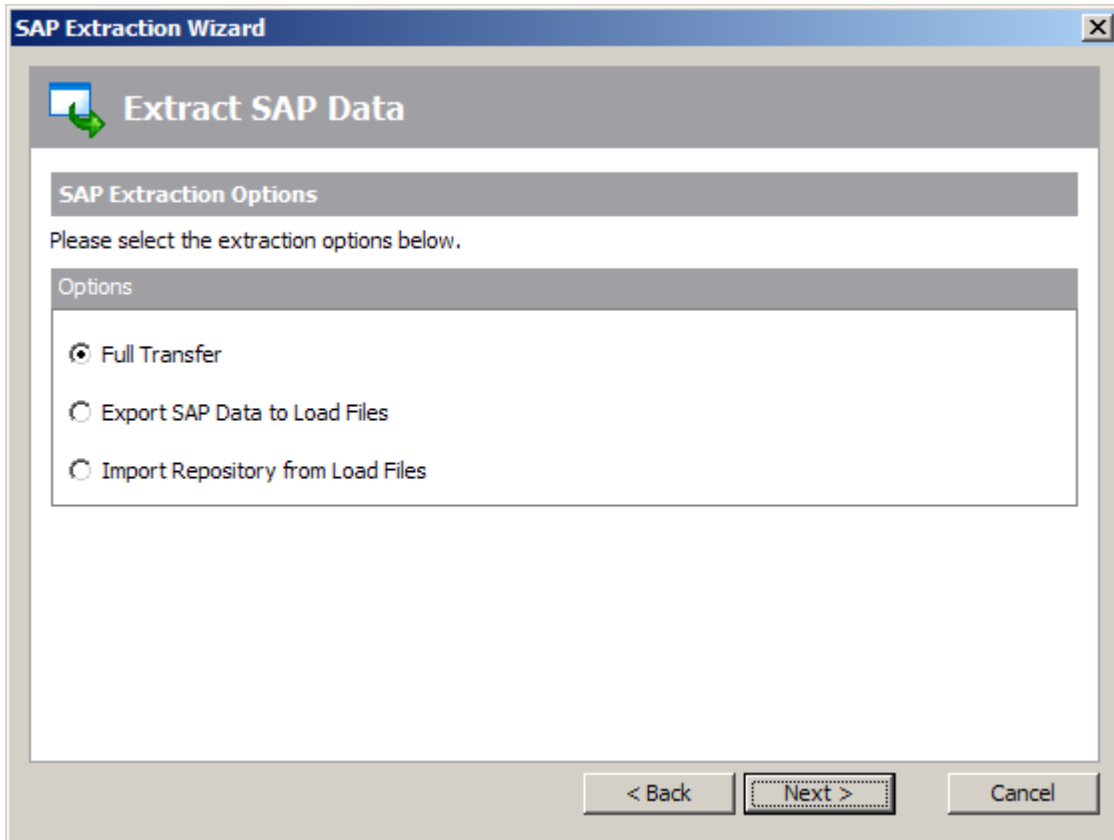


SAP Extraction Wizard – Extraction Languages

SAP holds metadata in a wide range of languages. This means that you can find descriptive table and column names, plus attribute definitions in English, German, French, Italian and many other languages. The Saphir extraction process lets you extract metadata for one or more of these languages.

On the Extraction Languages form, check each of the language codes you require (you must select at least one). If you do not see the languages you require, you can extend the list by editing the SaphirSettings.xls file. See Appendix B for details of this file.

Now click the 'Next' button to proceed to the next stage of the Extraction Wizard. If the metadata extraction for SAP is to be performed by the RFC Connection method, then the next form to be displayed will be the 'SAP Extraction Options' form. The purpose of this form is described on the next page. If the Direct DB Connect method has been chosen instead, you can skip to the next Extraction Wizard form.



SAP Extraction Wizard – SAP Extraction Options

The RFC Connect extraction method extracts metadata from SAP by invoking an SAP ABAP function which writes out a series of text files for each of the SAP metadata tables being interrogated. These text files are read by Saphir and the contents used to load the Saphir repository with the extracted data. The SAP Extraction Options form enables this process to be split into two discreet phases, should you wish. The ‘Options’ radio button controls this process. The possible values are:

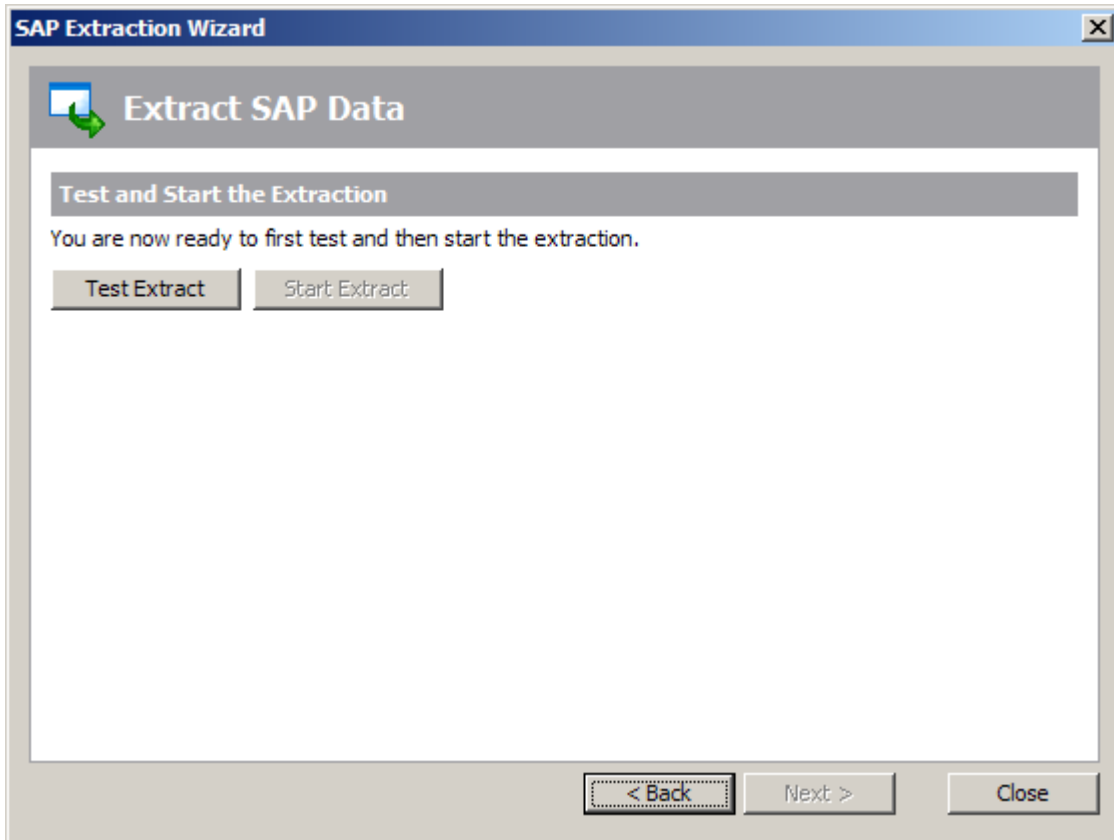
*Full Transfer*     The metadata is exported to text files and then immediately read into the Saphir repository

*Export SAP Data to Load Files*     The metadata is exported to text files only. The Saphir repository is not updated.

*Import Repository from Load Files*     Existing text files, created by a previous ‘Export SAP Data to Load Files’ session, are used to populate the Saphir repository

### 3.1.1 Why would I use the Export/Import options?

The most likely use these options is when the person requiring the metadata has no access to the source SAP system and requires the extraction to be done by someone else. We recommend that you choose the ‘Full Transfer’ extraction option if you have no real reason to split the extraction process into two stages.



SAP Extraction Wizard – Test and Start the Extraction

This form is the last stage of the Extraction Wizard process.

Clicking the 'Test Extract' button causes Saphir to run a series of queries to ensure that all the objects requested for extraction are available.

Assuming the 'Text Extract' was successful, click the 'Start Extract' button to begin the full extraction process.

A series of status messages will allow you to monitor the extraction process. Once the extraction has completed, you will be ready to browse the extracted metadata. An overview of the Saphir features for achieving this is provided in the next chapter.

**Test and Start the Extraction**

You are now ready to first test and then start the extraction.

Extraction Status

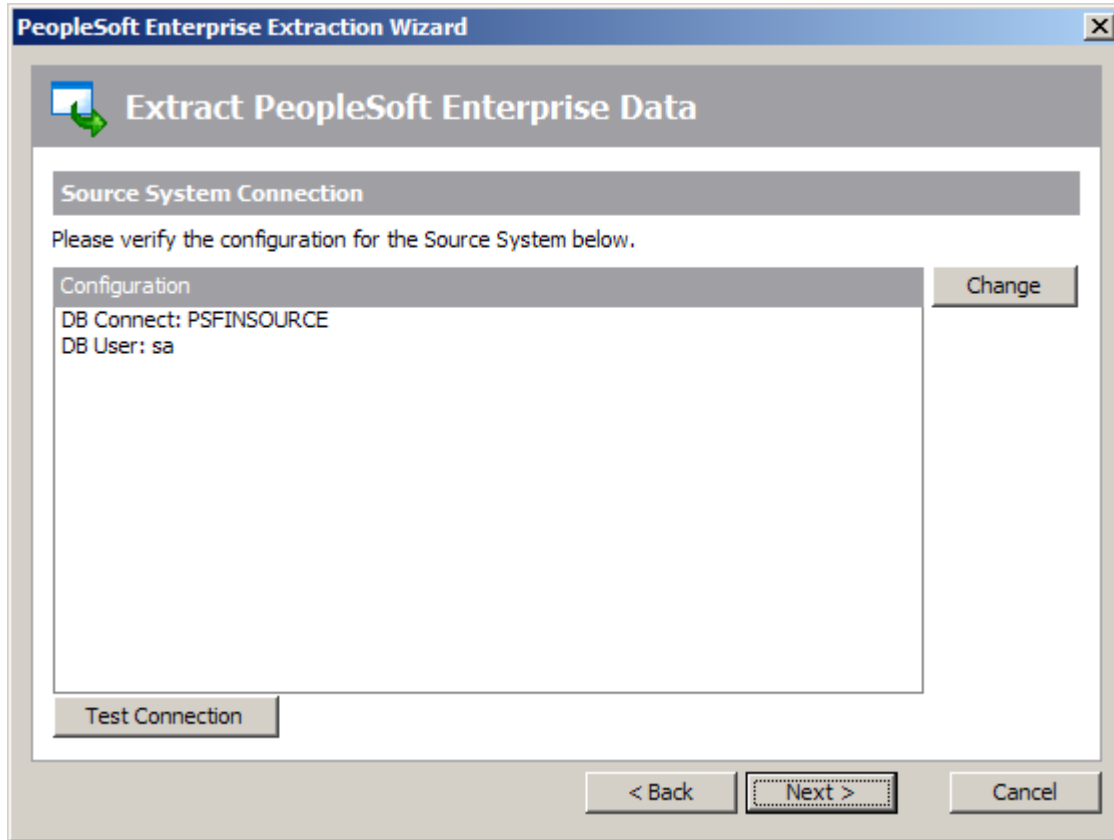
Repository Reorganisation process [Sync DD\_Field DomainName]  
Reorganisation ...

- [-] ▶ PeopleSoft Enterprise Repository Extraction (Mode=Run) 11:11:47
  - [+] ✓ Saphir-Environment 11:11:47 !Msg!
  - [+] ✓ Repository: PSFRANCEV5 11:11:47 !Msg!
  - [+] ✓ Check PeopleSoft Enterprise Environment 11:11:47 [00:00:03]
  - [+] ✓ prepare/clear affected repository tables 11:11:51 [00:00:18]
  - [+] ✓ DD\_Languages / Fill Languages 11:12:10
  - [+] ✓ Initialize Language Environment from DD\_Languages 11:12:10
  - [+] ✓ Domain 11:12:10 [00:00:01]
  - [+] ✓ DataElement 11:12:11 [00:00:02]
  - [+] ✓ DataElement Description (default language) 11:12:14 [00:00:02]
  - [+] ✓ DataElement Description (additonal languages) 11:12:16
  - [+] ✓ Table 11:12:16
  - [+] ✓ Table Description (default language) 11:12:17
  - [+] ✓ Table Description (additional languages) 11:12:17
  - [+] ✓ Check of DD\_Domain / check if DD\_Lockup\_EntityName do exist 11:12:17
  - [+] ✓ Fields and Indices 11:12:17 [00:00:17]
  - [-] ▶ Sync DD\_Fields / get DD\_DomainNames from DD\_DataElements 11:12:35
    - [-] ▶ Repository Reorganisation process [Sync DD\_Field DomainName] 11:12:35
      - ✓ DD\_FIELD: Synchronize DomainName 11:12:35

Extraction log during the extraction process

### 3.2 Extracting metadata from PeopleSoft Enterprise

Start the extraction process by selecting the 'ERP Extract' option from the Saphir File menu. This will start the PeopleSoft Extraction Wizard. Click the 'Next' button on the first 'Extraction Wizard' form to display the 'Source System Connection' form.



PeopleSoft Extraction Wizard – Source System Connection

This form shows a summary of the connection parameters, previously supplied when completing the Repository Wizard. You can:

- Click the 'Change' button to further amend the connection parameters
- Click the 'Test Connection' button to verify the connection through to the Application.

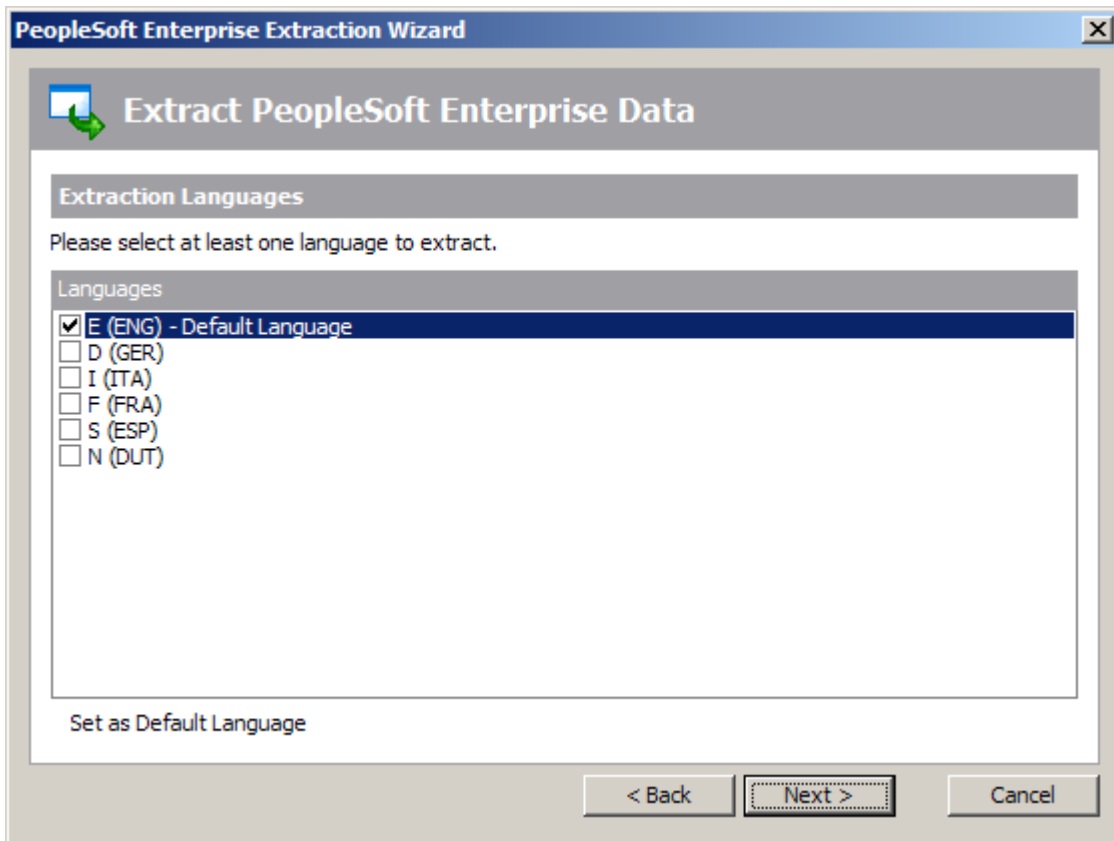
Now click the 'Next' button to proceed to the next stage.

Steps	
<input checked="" type="checkbox"/>	Log Source DB Physical Connection
<input checked="" type="checkbox"/>	Check PeopleSoft Enterprise Environment
<input checked="" type="checkbox"/>	Clear affected repository tables or recreate the whole repository
<input checked="" type="checkbox"/>	Log Repository DB Physical Connection
<input checked="" type="checkbox"/>	Prepare the Repository Environment (after deleting unused data)
<input checked="" type="checkbox"/>	DD_LANGUAGES / Fill Languages
<input checked="" type="checkbox"/>	Initialize Language Environment from DD_LANGUAGES
<input checked="" type="checkbox"/>	Domain
<input checked="" type="checkbox"/>	DataElement
<input checked="" type="checkbox"/>	DataElement Description (default language)
<input checked="" type="checkbox"/>	DataElement Description (additional languages)
<input checked="" type="checkbox"/>	Table
<input checked="" type="checkbox"/>	Table Description (default language)
<input checked="" type="checkbox"/>	Table Description (additional languages)
<input checked="" type="checkbox"/>	Check of DD_DOMAIN / check if DD_LOOKUP_ENTITYNAME exists
<input checked="" type="checkbox"/>	Fields and Indices
<input checked="" type="checkbox"/>	Sync DD_FIELD / get DD_DOMAINNAMES from DD_DATAELEMENTS
<input checked="" type="checkbox"/>	Application Hierarchy
<input checked="" type="checkbox"/>	Delete DD_TABLES without Fields
<input checked="" type="checkbox"/>	Sync DD_FIELD Key Flags with Primary Key IndexMembers
<input checked="" type="checkbox"/>	Reorganisation of Repository after Extract
<input checked="" type="checkbox"/>	Generate ERP Based Relationships
<input type="checkbox"/>	Generate Rule Based Relationships
<input type="checkbox"/>	Generate Extended Relationships

PeopleSoft Extraction Wizard – Extraction Steps

This form lists each of the stages involved in extracting metadata from a PeopleSoft system. The first 22 steps are checked ‘On’ by default. The 23<sup>rd</sup> and 24<sup>th</sup> steps (Generate Rule Based Relationships and ‘Generate Extended Relationships’) can be used to create additional relationships between PeopleSoft tables using user-specified rules and Saphir inference rules respectively. We recommend that you only select either of the last two options after fully understanding the processes described in Appendix B(‘Adding Additional Relationships’) of the ‘Saphir User Guide.’

Click the ‘Next’ button to proceed to the next stage’.

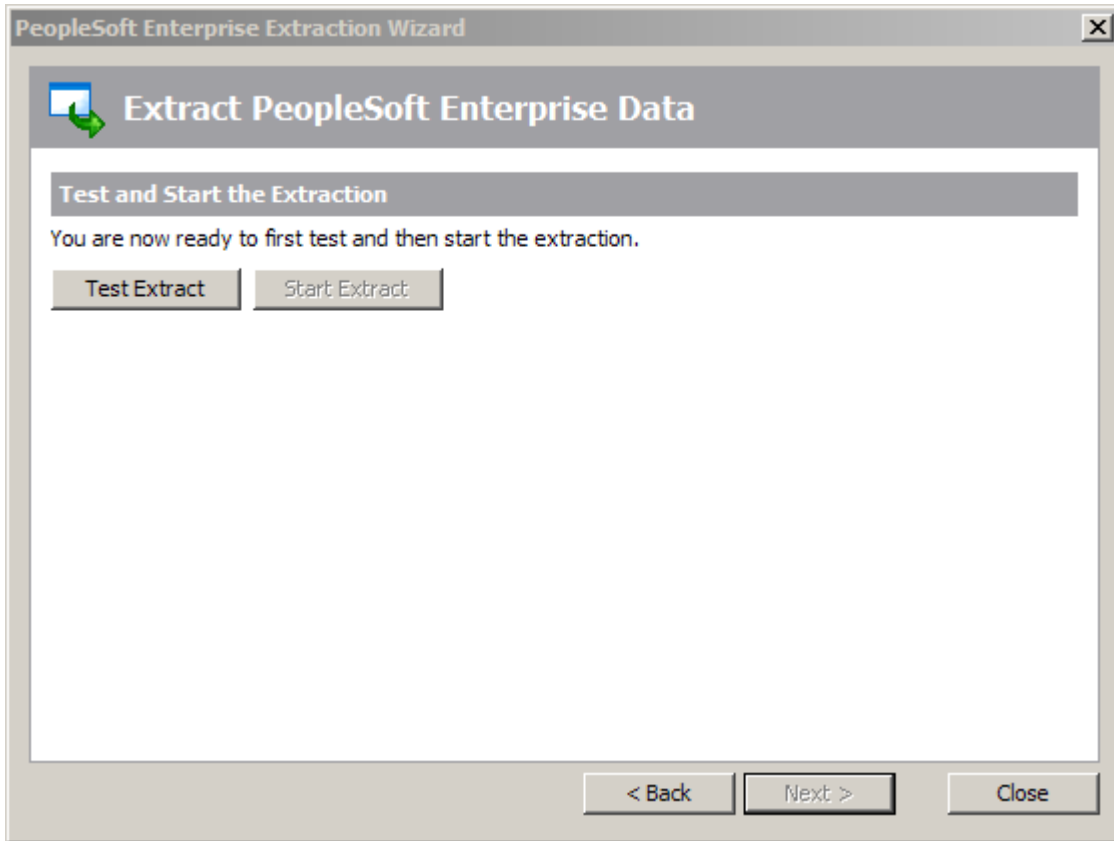


PeopleSoft Extraction Wizard – Extraction Languages

PeopleSoft holds metadata in a wide range of languages. This means that you can find descriptive table and column names, plus attribute definitions in English, German, French, Italian and many other languages. The Saphir extraction process lets you extract metadata for one or more of these languages.

On the Extraction Languages form, check each of the language codes you require (you must select at least one). If you do not see the languages you require, you can extend the list by editing the Saphir.ini file. See Appendix B for details of this file.

Now click the 'Next' button to proceed to the next stage of the Extraction Wizard.



PeopleSoft Extraction Wizard – Test and Start the Extraction

This form is the last stage of the Extraction Wizard process.

Clicking the ‘Test Extract’ button causes Saphir to run a series of queries to ensure that all the objects requested for extraction are available.

Assuming the ‘Text Extract’ was successful, click the ‘Start Extract’ button to begin the full extraction process.

A series of status messages will allow you to monitor the extraction process. Once the extraction has completed, you will be ready to browse the extracted metadata. An overview of the Saphir features for achieving this is provided in the next chapter.

### 3.3 Extracting metadata from Siebel

Start the extraction process by selecting the 'ERP Extract' option from the Saphir File menu. This will start the Siebel Extraction Wizard. Click the 'Next' button on the first 'Extraction Wizard' form to display the 'Source System Connection' form.

Siebel Extraction Wizard – Source System Connection

This form shows a summary of the connection parameters, previously supplied when completing the Repository Wizard. You can:

- Click the 'Change' button to further amend the connection parameters
- Click the 'Test Connection' button to verify the connection through to the Application.

Now click the 'Next' button to proceed to the next stage.

Steps
<input checked="" type="checkbox"/> Check existence of Siebel Repository ID
<input checked="" type="checkbox"/> prepare/clear affected repository tables
<input checked="" type="checkbox"/> DD_Languages / Fill Languages
<input checked="" type="checkbox"/> Domain
<input checked="" type="checkbox"/> DomainValue
<input checked="" type="checkbox"/> DataElement
<input checked="" type="checkbox"/> Table
<input checked="" type="checkbox"/> Fields + Primary Keys
<input checked="" type="checkbox"/> Index
<input checked="" type="checkbox"/> Relationships
<input checked="" type="checkbox"/> View
<input checked="" type="checkbox"/> ViewField
<input checked="" type="checkbox"/> Application + Business Object Hierarchy
<input checked="" type="checkbox"/> Application + Screen + View Hierarchy
<input checked="" type="checkbox"/> Application + Screen + View Hierarchy / Step 2
<input checked="" type="checkbox"/> Delete DD_Views without Views
<input checked="" type="checkbox"/> Reorganisation of Repository after Extract

Siebel Extraction Wizard – Extraction Steps

This form lists each of the stages involved in extracting metadata from a Siebel system. The 17 steps are checked 'On' and we recommend that you leave this selection and continue to the next stage.

Click the 'Next' button to proceed to the next stage.

**Extract Siebel Data**

**Extraction Languages**

Please select at least one language to extract.

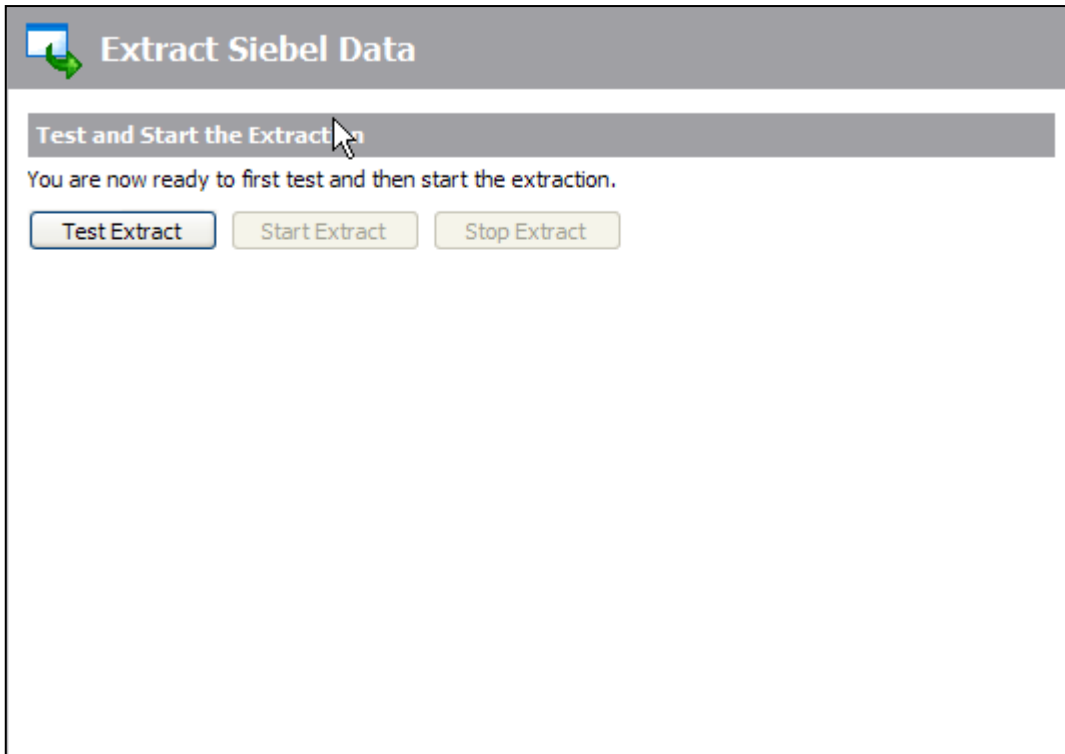
Languages

- E (E) - Default Language
- D (D)

Siebel Extraction Wizard – Extraction Languages

On the Extraction Languages form, check the language code you require.

Now click the 'Next' button to proceed to the next stage of the Extraction Wizard.



Siebel Extraction Wizard – Test and Start the Extraction

This form is the last stage of the Extraction Wizard process.

Clicking the ‘Test Extract’ button causes Saphir to run a series of queries to ensure that all the objects requested for extraction are available.

Assuming the ‘Text Extract’ was successful, click the ‘Start Extract’ button to begin the full extraction process.

A series of status messages will allow you to monitor the extraction process. Once the extraction has completed, you will be ready to browse the extracted metadata. An overview of the Saphir features for achieving this is provided in the next chapter.

### 3.4 Extracting metadata from J.D.Edwards EnterpriseOne

Start the extraction process by selecting the 'ERP Extract' option from the Saphir File menu. This will start the EnterpriseOne Extraction Wizard. Click the 'Next' button on the first 'Extraction Wizard' form to display the 'Source System Connection' form.

**JD Edwards EnterpriseOne Extraction Wizard**

**Extract JD Edwards EnterpriseOne Data**

**Source System Connection**

Please verify the configuration for the Source System below.

Configuration	Change
DB Connect: JDELOCAL	Change
DB User: sa	
JDE EntOne XML Directory: C:\erpdata\jde9	

Test Connection

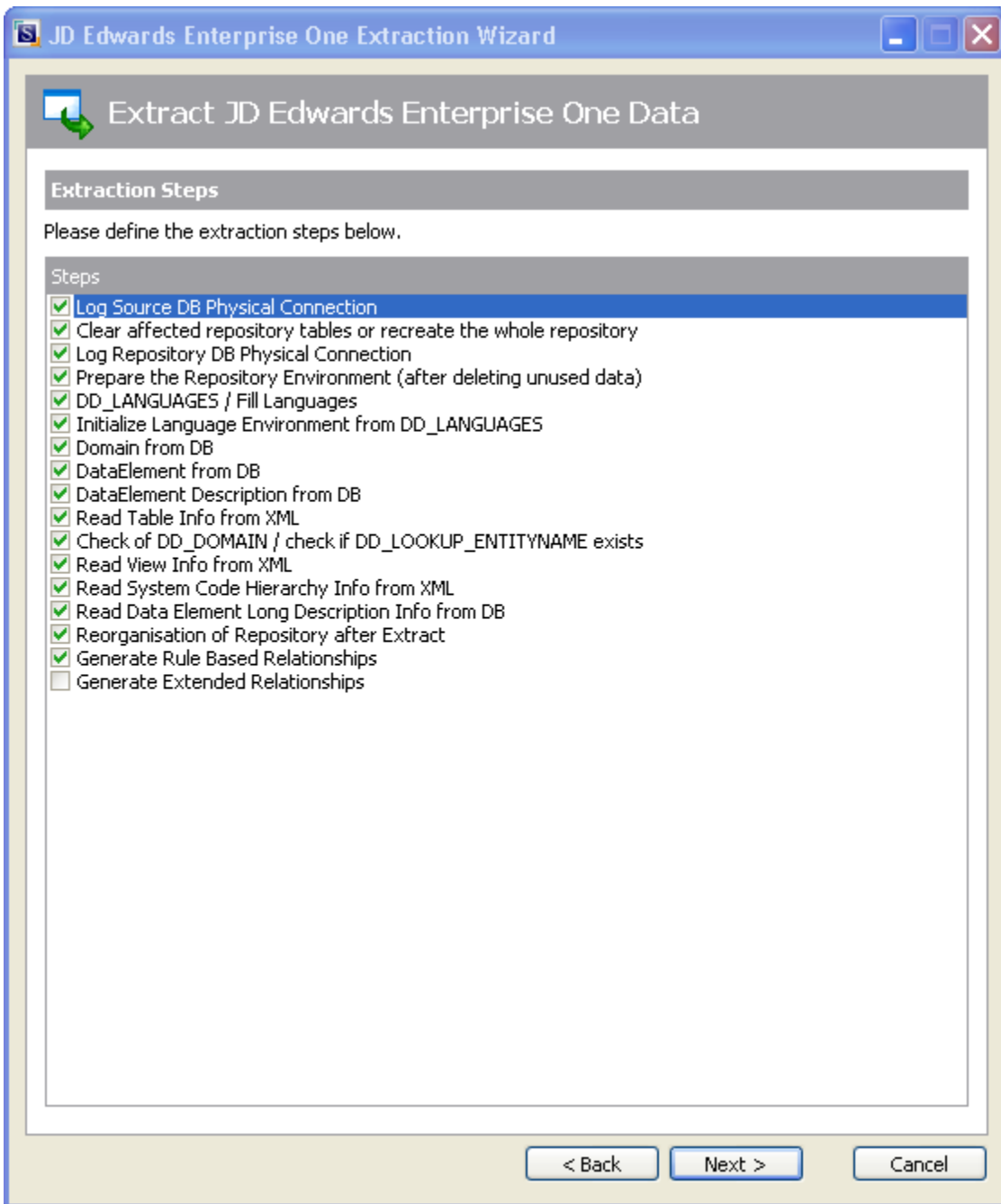
< Back    Next >    Cancel

J.D.Edwards EnterpriseOne Extraction Wizard – Source System Connection

This form shows a summary of the connection parameters, previously supplied when completing the Repository Wizard. You can:

- Click the 'Change' button to further amend the connection parameters
- Click the 'Test Connection' button to verify the connection through to the Application.

Now click the 'Next' button to proceed to the next stage.

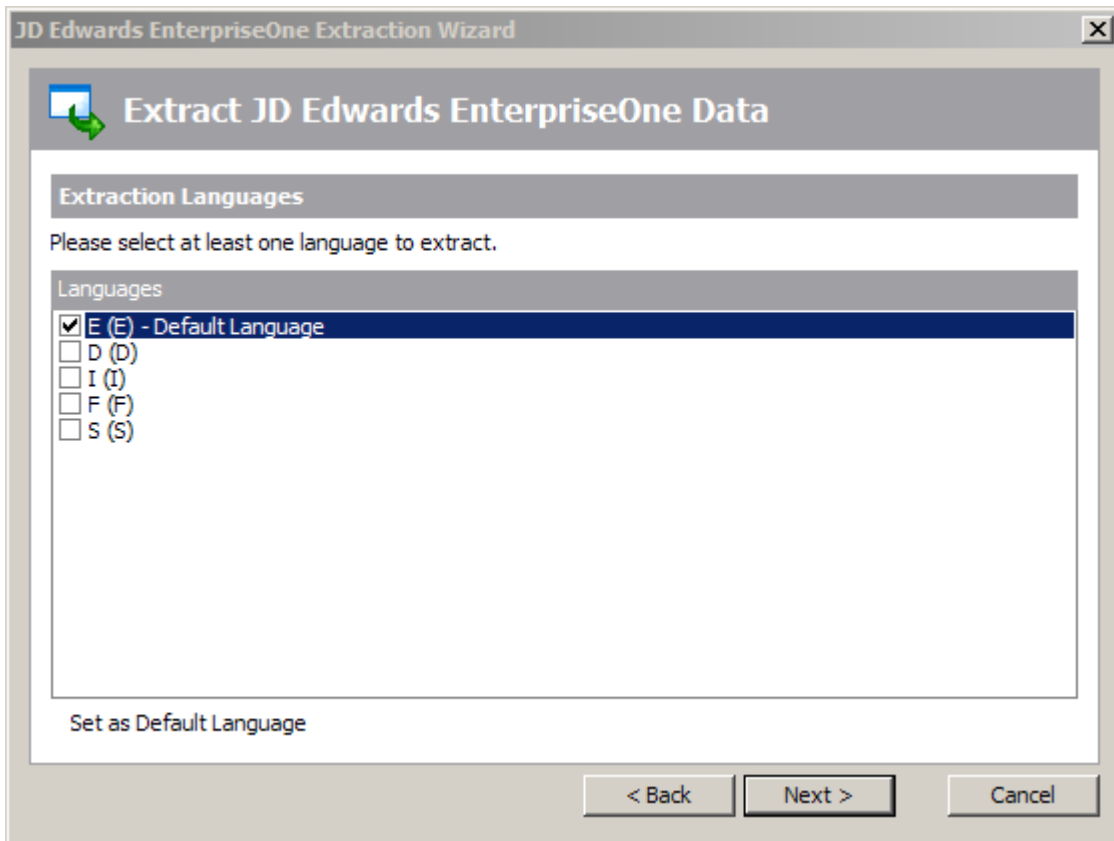


J.D.Edwards EnterpriseOne Extraction Wizard – Extraction Steps

This form lists each of the stages involved in extracting metadata from an EnterpriseOne system.

The 16<sup>th</sup> and 17<sup>th</sup> steps (Generate Rule Based Relationships and ‘Generate Extended Relationships’) can be used to create additional relationships between JDedwards tables using user-specified rules and Saphir inference rules respectively. We recommend that you only select either of the last two options after fully understanding the processes described in Appendix B (‘Adding Additional Relationships’) of the ‘Saphir User Guide.’

Click the ‘Next’ button to proceed to the next stage’.

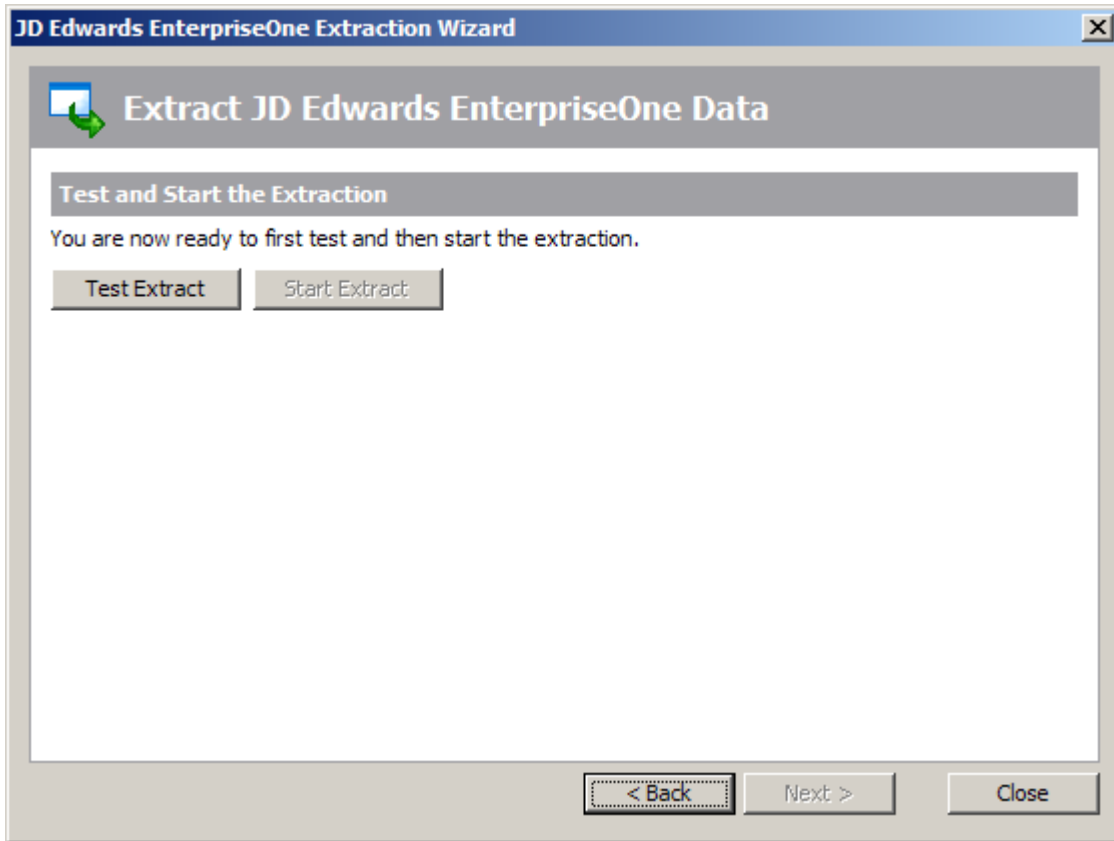


EnterpriseOne Extraction Wizard – Extraction Languages

EnterpriseOne holds metadata in a range of languages. This means that you can find descriptive table and column names, plus attribute definitions in English, German, French, Italian and many other languages. The Saphir extraction process lets you extract metadata for one or more of these languages.

On the Extraction Languages form, check each of the language codes you require (you must select at least one). If you do not see the languages you require, you can extend the list by editing the SaphirSettings.xls file. See Appendix B for details.

Now click the 'Next' button to proceed to the next stage of the Extraction Wizard.



EnterpriseOne Extraction Wizard – Test and Start the Extraction

This form is the last stage of the Extraction Wizard process.

Clicking the ‘Test Extract’ button causes Saphir to run a series of queries to ensure that all the objects requested for extraction are available.

Assuming the ‘Text Extract’ was successful, click the ‘Start Extract’ button to begin the full extraction process.

A series of status messages will allow you to monitor the extraction process. Once the extraction has completed, you will be ready to browse the extracted metadata. An overview of the Saphir features for achieving this is provided in the next chapter.

### 3.5 Extracting metadata from Oracle EBS

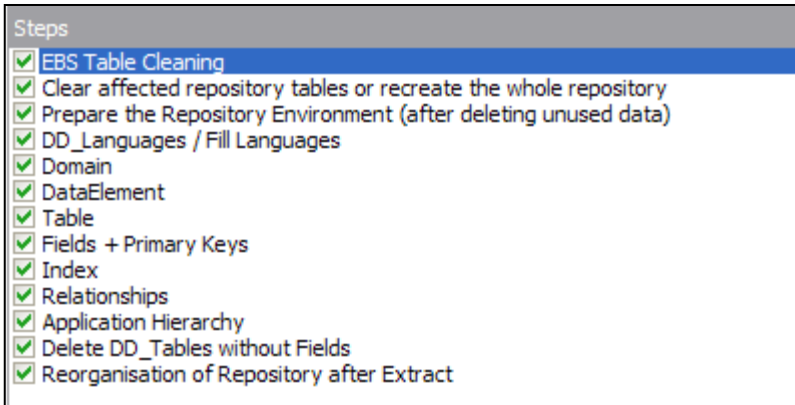
Start the extraction process by selecting the 'ERP Extract' option from the Saphir File menu. This will start the EBS Extraction Wizard. Click the 'Next' button on the first 'Extraction Wizard' form to display the 'Source System Connection' form.

Oracle EBS Extraction Wizard – Source System Connection

This form shows a summary of the connection parameters, previously supplied when completing the Repository Wizard. You can:

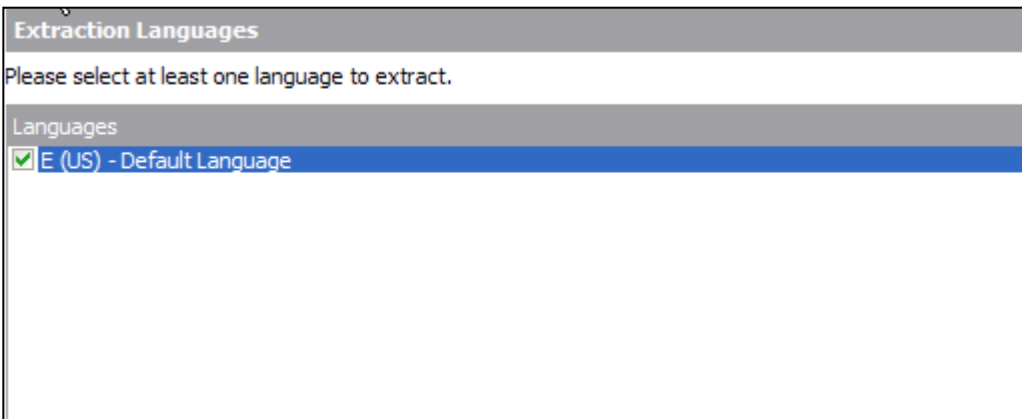
- Click the 'Change' button to further amend the connection parameters
- Click the 'Test Connection' button to verify the connection through to the Application.

Now click the 'Next' button to proceed to the next stage.



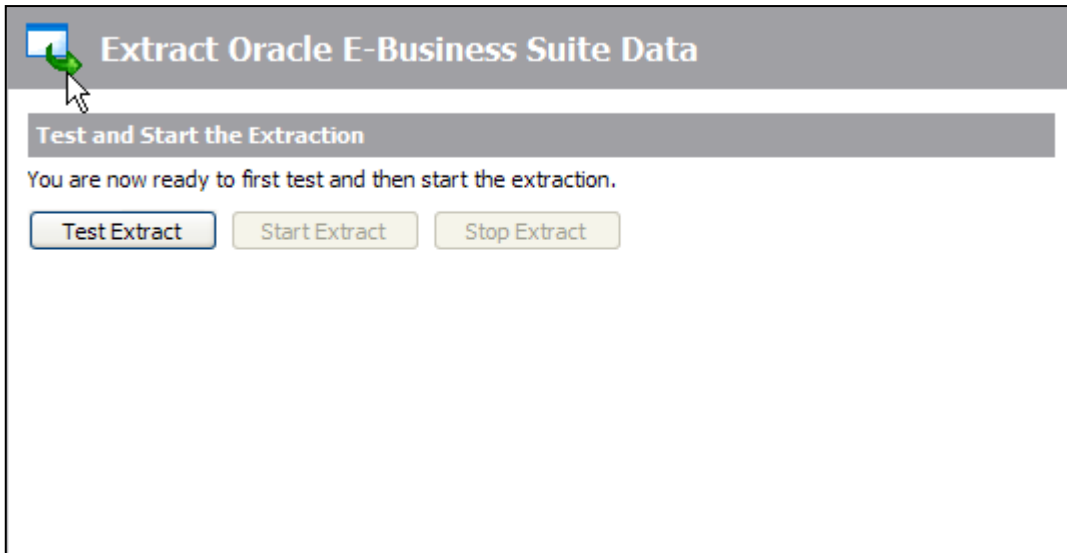
EBS Extraction Wizard – Extraction Steps

This form lists each of the stages involved in extracting metadata from an EBS. The 13 steps are checked 'On' and we recommend that you leave this selection and then click the 'Next' button to proceed to the next stage'.



EBS Extraction Wizard – Extraction Languages

The extraction of metadata from EBS will always be in US English. Click the 'Next' button to proceed to the next stage of the Extraction Wizard.



EBS Extraction Wizard – Test and Start the Extraction

This form is the last stage of the Extraction Wizard process.

Clicking the 'Test Extract' button causes Saphir to run a series of queries to ensure that all the objects requested for extraction are available.

Assuming the 'Text Extract' was successful, click the 'Start Extract' button to begin the full extraction process.

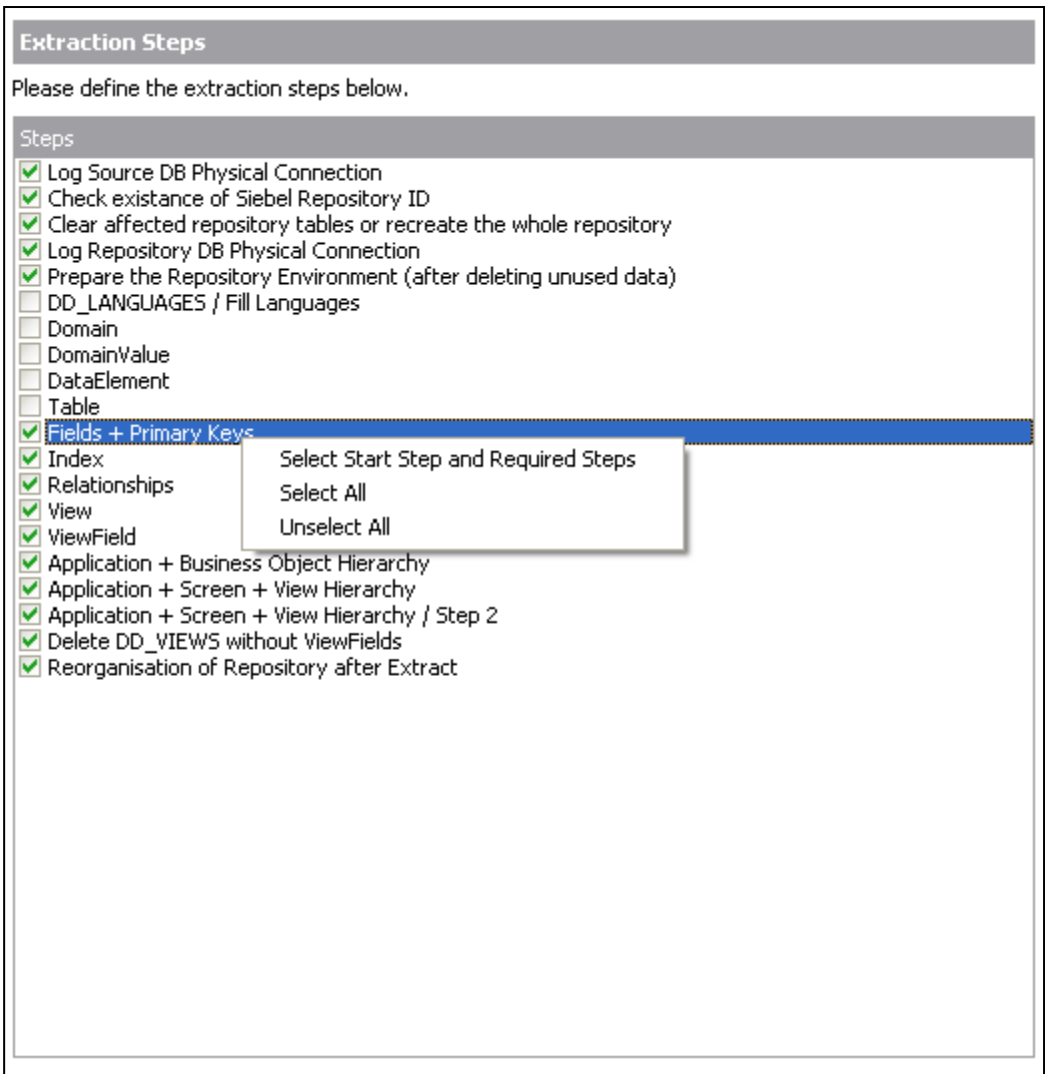
A series of status messages will allow you to monitor the extraction process. Once the extraction has completed, you will be ready to browse the extracted metadata. An overview of the Saphir features for achieving this is provided in the next chapter.

### 3.6 Restarting an extraction

There may be circumstances where an extraction fails or has to be abandoned. In such a case, it is possible to restart the extraction process.

To achieve this, follow the steps described above for the required source ERP system until the Extraction Steps screen. From this screen, click on the step which is the first step to be performed and Right Mouse Click to show the RMC menu. Now select the ‘Select Start Step and Required Steps’.

This will select any steps that need to be performed, plus the subsequent required steps. Then proceed with the extraction as normal.



Selecting the restart point from the Extraction Steps screen


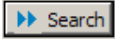
## 4 Quick Tour of Saphir's Features

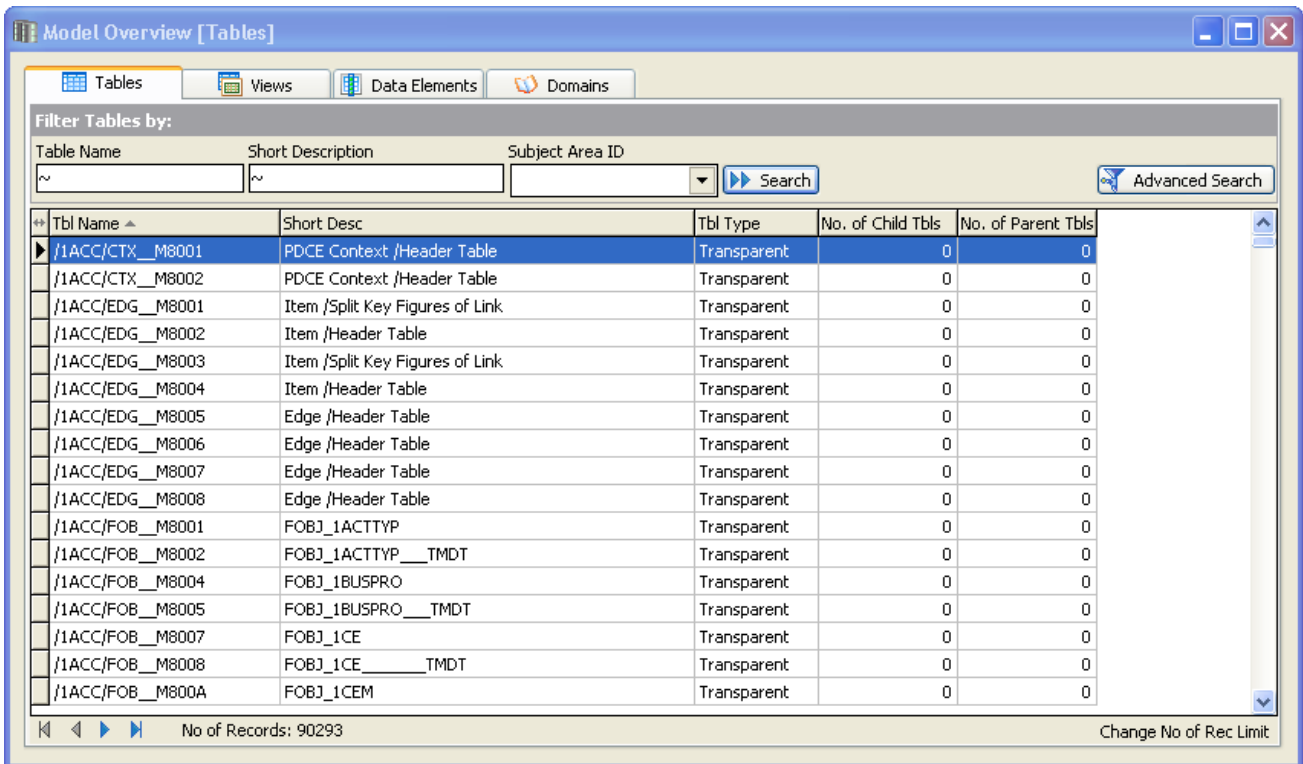
*Gives an overview of the main Saphir features for browsing the extracted metadata.*

The extraction of metadata from your chosen Enterprise Application was described in the previous chapter. Once this process is complete, Saphir provides a convenient interface for exploring the metadata and exporting sub-sets into other environments.

### 4.1 Model Overview

The Saphir 'Model Overview' is a good place to start exploring the metadata extracted from your application, such as searching for tables, their characteristics and how they are related.

Click  on the Saphir toolbar to display the 'Model Overview' form. Then clicking the  button will return a list of all the tables in the Saphir repository, as shown below, which can be scrolled through.

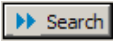


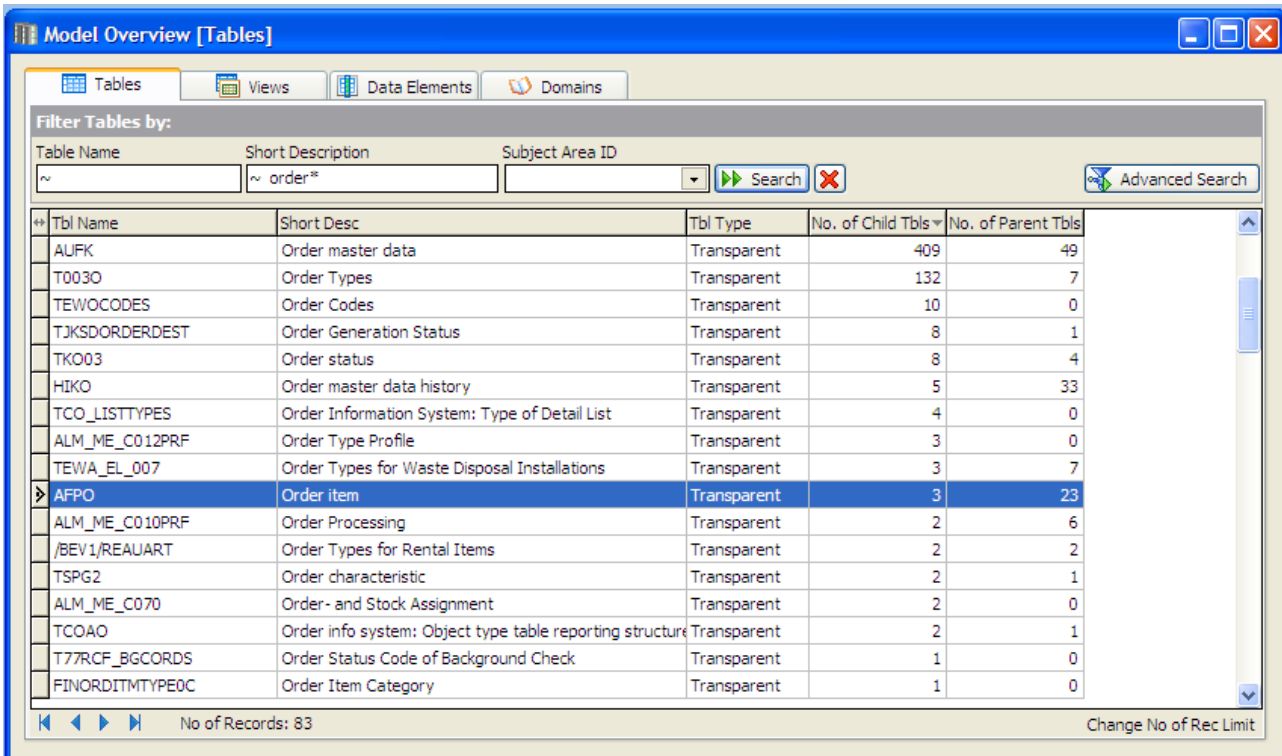
The Saphir Model Overview

#### 4.1.1 Tables

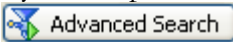
The fields displayed for each table, in the 'Model Overview' include the internal table name 'Tbl Name' and some table descriptive text 'Short Desc'. The 'No. of Child Tbls' and 'No. of Parent Tbls' show the number of child and parent tables related to each individual table. The total number of tables or records is shown at the bottom of the frame.

Use the 'Filter Tables by': section in the upper part of the screen to search for tables with certain criteria and reduce the list of tables, or find a specific table. For example, type '\*Order\*' into the 'Short Description' box

and click . This will return a list of all tables with the characters 'Order' in the 'Short Desc' field. The 'Number of Records:' at the bottom of the screen reduces.

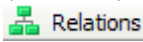

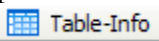


Model Overview showing the results of a Search

The total number of tables extracted from a full ERP database into the Saphir repository is usually very large, e.g. over 90,000 shown in the example above, too many to understand and manage easily as a single group. So an important facility provided by Saphir is its powerful filter/search capability to help the user find the tables appropriate to a particular requirement and then enable the tables to be grouped as required into manageable sets or subject areas for further analysis or export to another tool or diagrammer. See later section on **Filtering & Searching** including use of the  button and **Subject Areas**.

Clicking the  to the right of the  button is a quick way to clear all search criteria.

Double Click on one of the tables listed in the 'Model Overview' screen to see more information about the table highlighted. Information is shown in a series of tabs. The 'Fields' tab is introduced below.

More details are provided in the User Guide on the other tabs (Relations, Indexes, Table-Info). Click on the tab at the top of the screen to show these screens. The Relations tab  reveals a list of all the relationships between a table and any 'parent' tables. The Index tab  displays a list of indexes for the currently table. The Table-Info tab  shows details of the Table, it's Class and the Views based on the table.

### 4.1.2 Fields

The Fields tab shows a list of fields making up the table. It is the first tab to be presented of the Table tabs.

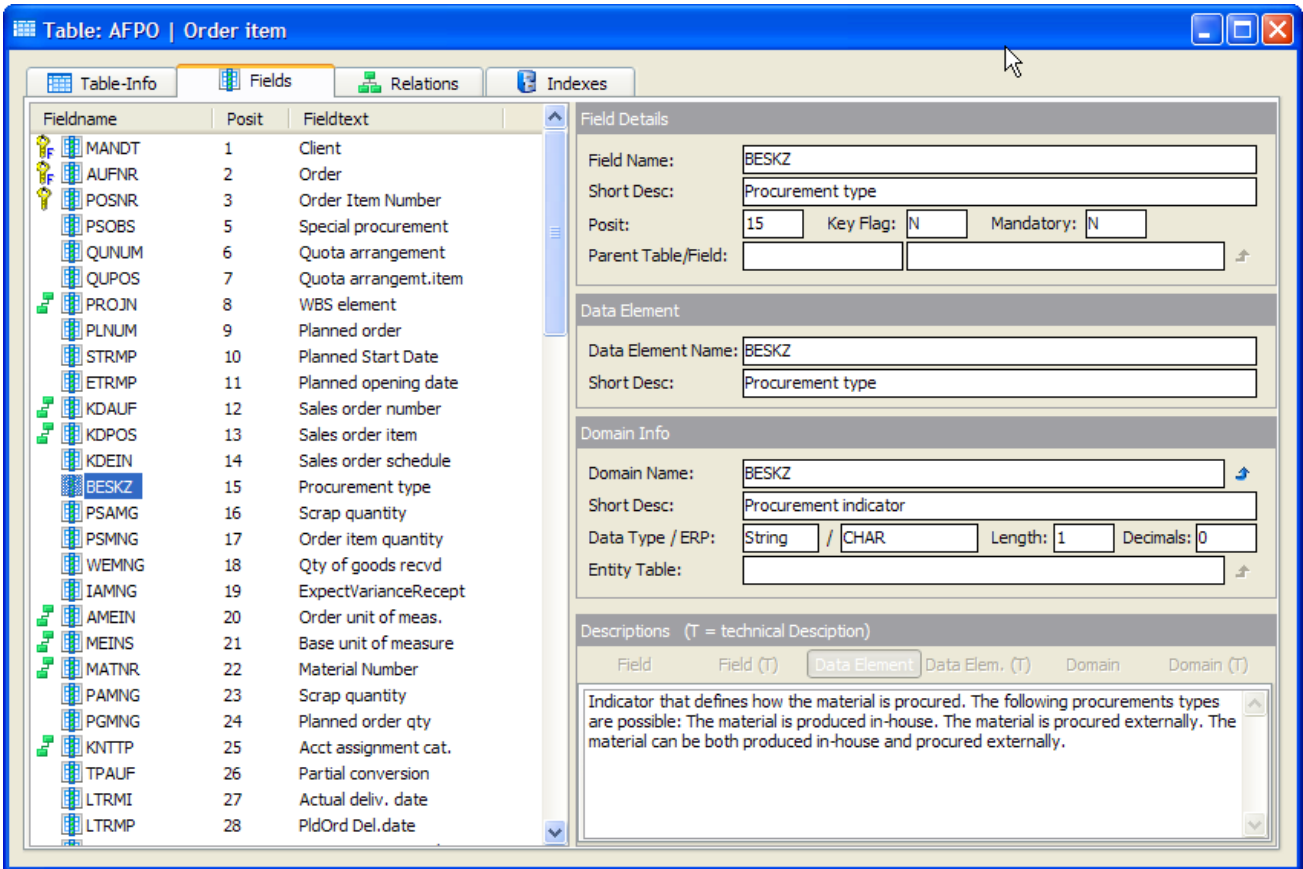






Table Field Details

Clicking on individual field names in the left hand side of the screen will cause the details about the field to be shown in the frames on the right hand side of the screen.

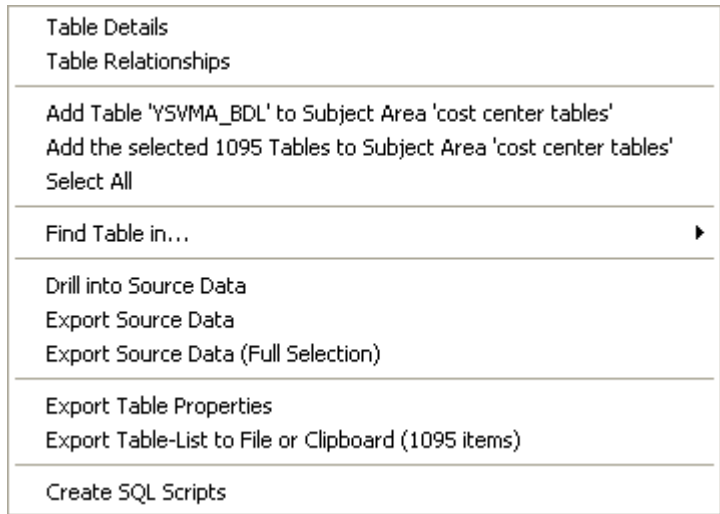
The default display format shows details of each field in the table. To the left of each field icon  there may be an additional icon. This can be: -

-  The Field is part of the Table's Primary Key
-  The Field is part of the Primary Key and also a Foreign Key
-  The Field is a Foreign Key field

### 4.1.3 Table Relationships

From the 'Tables' tab on the Model Overview Screen, select a table and click the Right mouse button. A number of options are presented.

Select 'Table Relationships' by moving the mouse over the option and clicking.

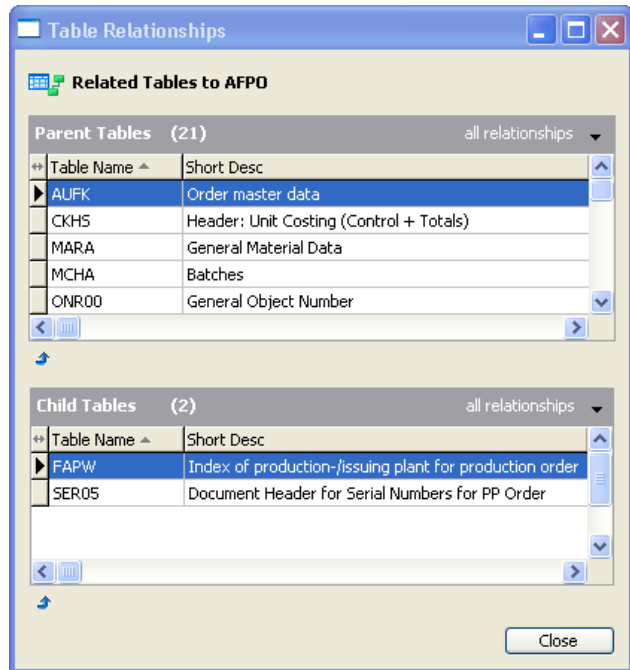


Model Overview Right Mouse Click menu


A new window is opened with details of the Tables linked to the selected table in the Model Overview.

The table name and description are shown of both the Parent and Child tables, making it easier to trace linked tables and include them in appropriate sets.


If a subject area is open then right mouse click menu will allow all or selected tables to be copied to the subject area.

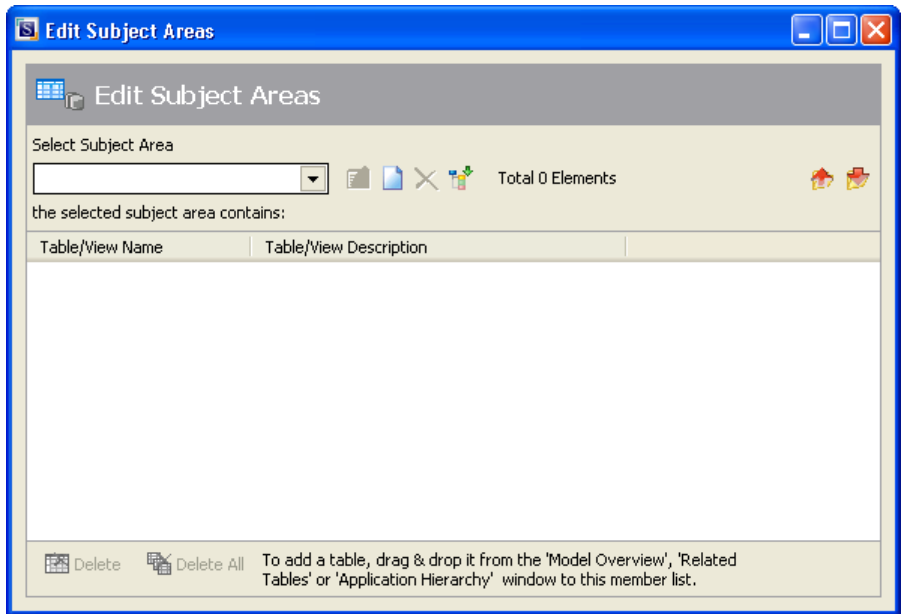


### 4.1.4 Filtering / Searching


The total number of tables extracted from a full ERP database into the Saphir repository is usually very large. So Saphir's powerful filter / search capability is essential to help the user find the tables appropriate to a particular requirement and enable the tables to be grouped as required into manageable sets or subject areas. This grouping ability makes it easier to understand and communicate appropriate parts of the metadata model. The concept of Subject Area groupings can also be utilised in filtering and searching, providing an additional dimension to limit the task to a particular set of tables. Criteria can be combined to allow more focused searches and the  **Advanced Search** button makes available a number of additional criteria to use. The Advanced Search capability, combined with the use of Subject Areas, also allows searches of one set of tables whilst excluding tables in another set.

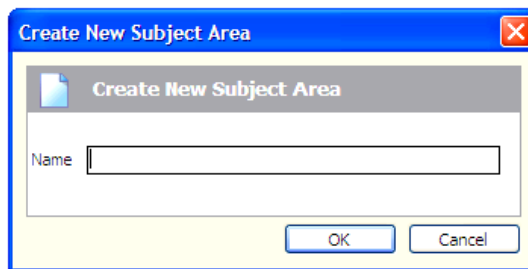
### 4.1.5 Subject Areas

To create and use a Subject Area, click the  icon on the Saphir toolbar. This will display the Edit Subject Areas form which has features for creating and populating subject areas.



The Subject Area form

Click the  icon to create a new subject area, enter a suitable name and click OK.

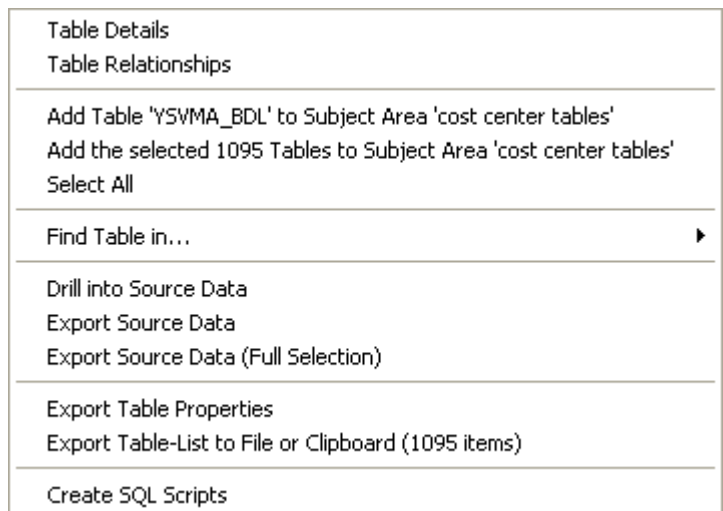


The next step is to add tables into the subject area.

Keep open the Subject Area into which you want to place tables.

Return to the Model Overview main 'Tables' tab. Select the required tables using a standard Shift/Click or Control/Click approach to choosing rows.

Then Right Mouse Click on one of the highlighted selected tables. The Right mouse Click menu displayed, as shown. Now choose the 'Add the selected...' option to add the selected tables to the currently open Subject Area.




Model Overview Right Mouse Click menu

There are other methods for populating a subject area, these are described fully in the Saphir User Guide.

### 4.1.6 Filtering using Subject Areas

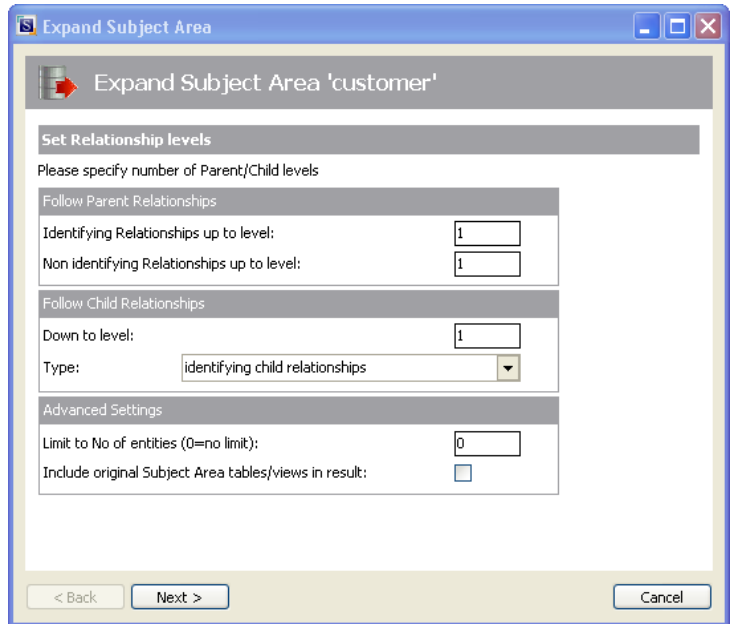
From the Model Overview screen, one of the filter criteria fields available is the Subject Area. This allows a Search to be focused on a set of tables included in the subject area. With this facility a chosen set of tables can be split into other sets or refined by using further searches. This is described further in the Saphir User Guide.

### 4.1.7 Expanding tables in a Subject Area

When there is one or more tables in the subject Area, the  icon on the subject area screen can be used to initiate the request for details of all tables linked to those in the subject area. This is a useful facility when tracing a particular area of the metamodel.

The number of levels of Parent and/or Child related tables to be assembled can be specified. Choosing just one level is often enough.

A list of the tables linked to those already in the subject area is presented. All or selected tables can then be added to the currently selected subject area, or a separate subject area.



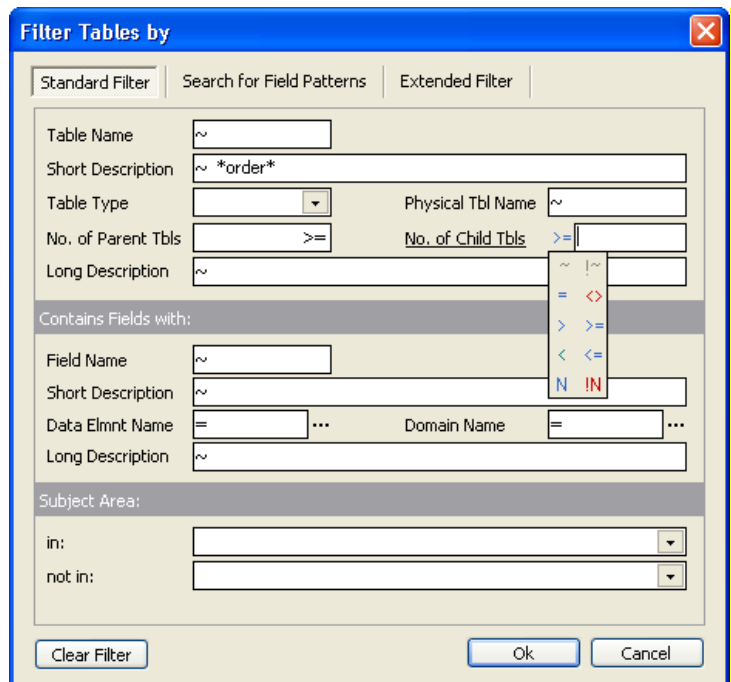
### 4.1.8 Advanced Search

From the Model Overview Tables screen. Click the Advance Search button to show the additional filter criteria available.

Filter criteria from the Model overview screen is carried forward to the Advanced Search.


The icons in the left or right hand portion of the fields allow more complex comparisons available. Click in the field to select the field and then click on the icon shown to the left of the field to select the appropriate criteria.

The section at the bottom of the Filter screen allows tables from a subject area to be included or excluded from a search.



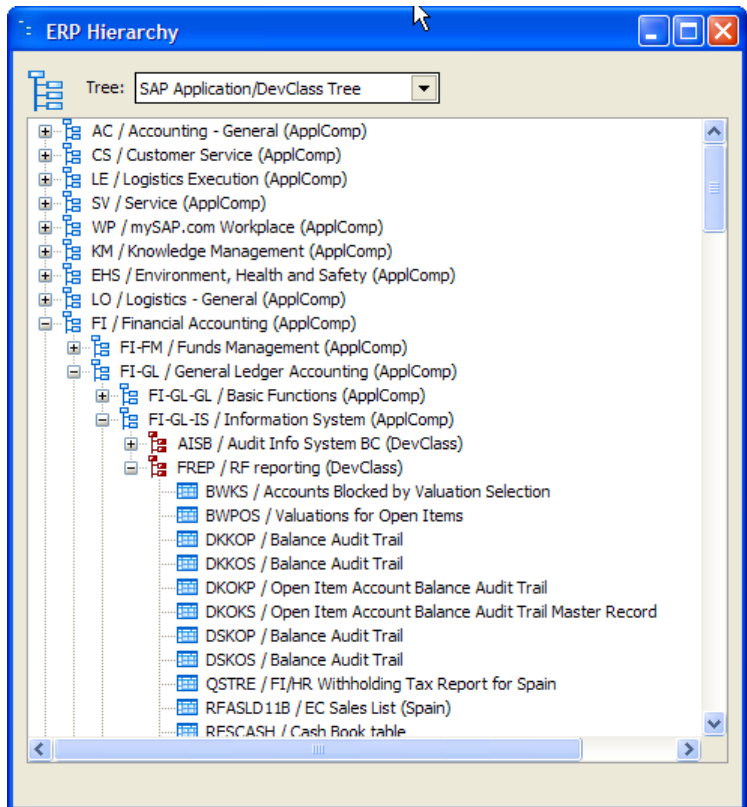
## 4.2 Navigating tables via the Application Hierarchy

The Application Hierarchy feature in Saphir is an alternative way of looking at the contents of the Saphir repository. The general concept of the Saphir 'hierarchy' feature is to show tables and/or views in logical groupings. The content of these hierarchies is very dependent on the ERP being used, but the general appearance and functionality is the same across all ERP types supported by Saphir.

Click the  icon on the Saphir toolbar to display the Application Hierarchy. This presents a list of modules and their associated tables. The actual content of the hierarchy shown will depend on the Enterprise Application being viewed. There may also be a choice of hierarchy, based on the drop down list at the top of the Application Hierarchy screen.


Clicking on a tree node expands a tree of subsidiary 'objects'. Right mouse clicking on an object with a subject area open provides a facility to add all objects of a given tree level to the selected subject area.

See the Saphir User Guide for more details on Application Hierarchy navigation.

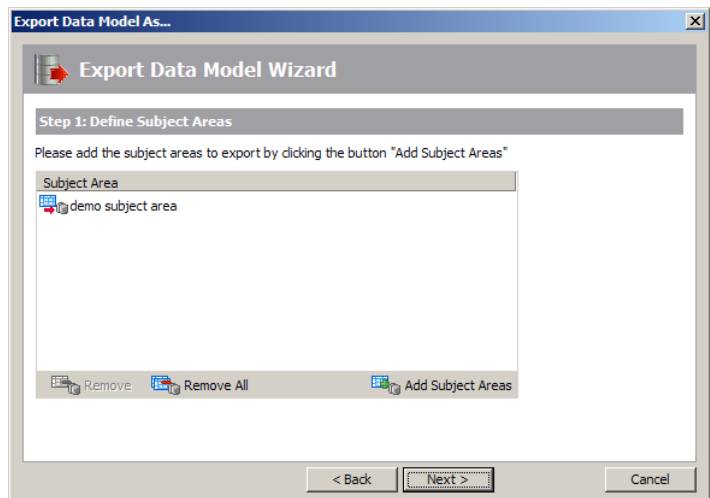


Displaying the Application Hierarchy

## 4.3 Exporting to third party tools

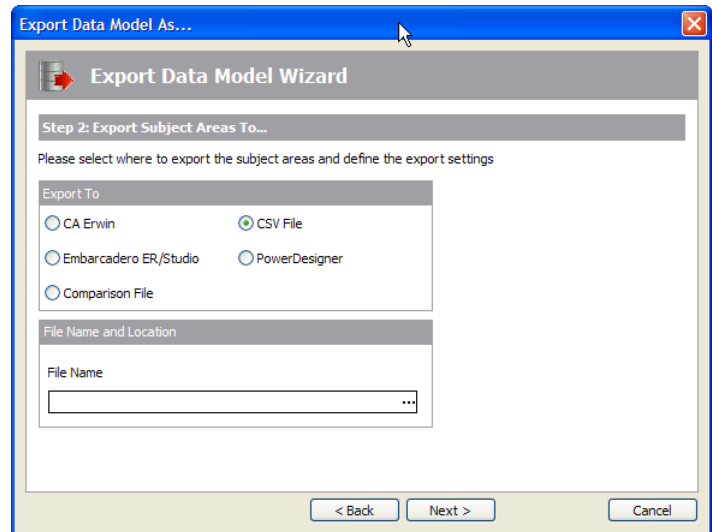
The Export to third party tools is based on the Saphir subject area concept. To begin the process of exporting the chosen data definitions to the tool of choice, click the  icon on the Saphir tool bar to open the Export Data Model Wizard. Click the 'next' button to start the export steps.

The 'Define Subject Areas' form is for selecting one or more subject areas to be exported. Click the 'Add Subject Areas' button and then use drag and drop to add one or more subject areas you require. Click the 'Next' button to progress to the next stage of the export wizard.



The Export Wizard – specifying the subject areas required

The next form displayed is the place to choose the export format. The actual export formats displayed and the subsequent wizard steps will depend on the formats that your Sapir installation has been configured to use.



Export Wizard – choosing the export format

## Appendix A. Installing the SAP ABAP Function

*Step-by-step guide to setting up the ABAP code used by Saphir to extract metadata*

If extraction of metadata from SAP or SAP BW via ABAP has been chosen, then the following steps will need to be completed by an SAP proficient programmer or administrator.

Saphir includes ABAP Functions which are run to extract metadata from the appropriate SAP data dictionary tables.

The steps below deal with configuring, loading and testing these functions.

The screen shots included in this section are from an SAP ECC 6 release – there may be slight differences in the screen layouts for earlier or later releases.

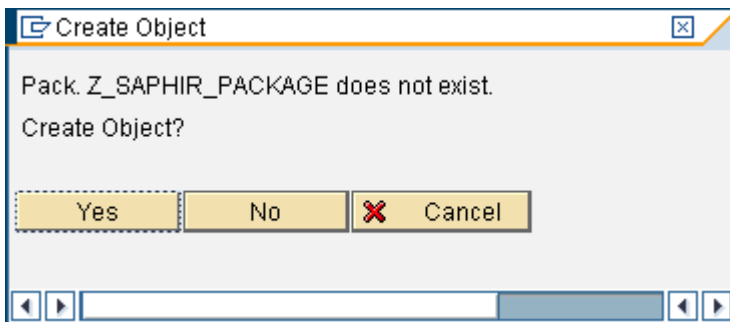
Note: Saphir is supplied with 2 additional ABAP functions (one each for SAP and SAP BW) that can be used to run the ‘main’ ABAP functions from within the SAP environment. These may be used to run the extraction process and create text files if it is not possible to access the SAP system via RFC calls. See Appendix E for details of installing and using these functions.

### A.1 Create a Package

(A Package is called a Development Class in earlier SAP releases).

An existing package can be used, or use these steps to create a new package. Go to the ABAP Workbench and proceed as follows:

1. Choose Object Navigator or call Transaction SE80
2. Choose ‘Package’ from the from down list
3. Enter a new package name (e.g. Z\_SAPHIR\_PACKAGE) and choose ‘Display’
4. Click ‘Yes’ on the ‘Create Object’ form to display the ‘Create Package’ form



Create Package Object

Package	Z_SAPHIR_PACKAGE
Short Description	Package for Saphir RFC use
Applic. Component	
Software Component	HOME
Transport Layer	
Package Type	Not a Main Package

Enter Package details

Enter a suitable 'Short Description' and click the Save icon.

## A.2 Create a Function Group

An existing function group can be used, or use the steps below to create a new group. This is a similar process to the creation of the Package.

From the Object Navigator, choose an object type of 'Function group'.

Enter a new function group name (e.g. Z\_SAPHIR\_RFC\_GROUP) and choose 'Display'.

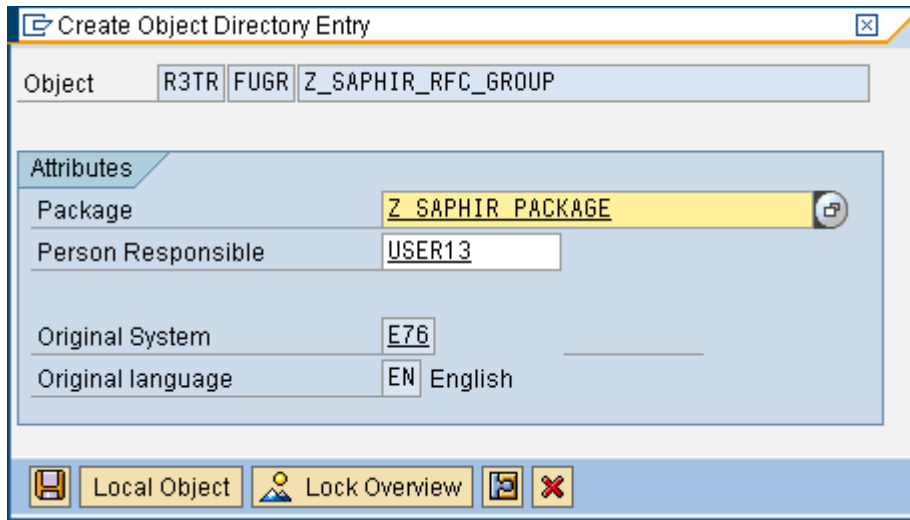
Click 'Yes' on the 'Create Object' form to display the 'Create Function Group' form.

Function group	Z_SAPHIR_RFC_GROUP
Short text	Function Group for Saphir
Person Responsible	USER13

Create Function Group

Enter a suitable 'Short Text' and click the Save icon.

This will prompt the display of the ‘Create Object Directory Entry’ form.



Create Object Directory Entry

Enter the name of the Package created in the previous step and click the save icon.

### A.3 Create the ABAP Function Module

To create the Function requires the one of the ABAP source code files that is delivered with Saphir. These files are installed in the ABAPFiles folder below the default Saphir folder and are as follows:

- For a SAP system: z\_saphir\_rfcs\_ddexpv5\_guidownload.txt.
- For a SAP BW system: z\_saphir\_rfcs\_bwexpv5\_guidownload.txt.

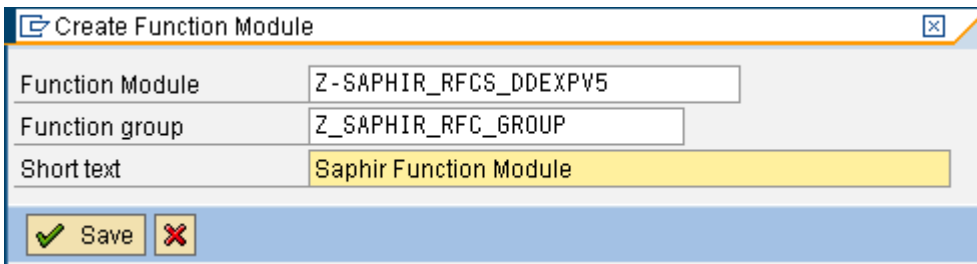
Open the appropriate delivered ABAP text file with Notepad and copy the full text of the file to the clipboard.

From the Object Navigator, right mouse click on the Function Group created in the previous step and click ‘Create’ and then choose ‘Function Module’.

On the ‘Create Function Module’ form, enter a name and a Short text description for the Module and click the Save icon. The name entered should reflect that of the chosen ABAP Function. That is:

For a SAP system: Z\_SAPHIR\_RFCS\_DDEXPV5

For a SAP BW system: Z\_SAPHIR\_RFCS\_BWEXPV5



Create the Function Module

The Function Module editor now appears. Click on the ‘Source code’ tab and replace the contents of the workspace with the contents of the clipboard copied from the appropriate ABAP function earlier.

```

1 | FUNCTION Z_SAPHIR_RFCS_DDEXPV5 .
2 | *-----
3 | *"Local interface:
4 | * IMPORTING
5 | *   VALUE(P_EXP_AREA) TYPE DDO2L-TABNAME
6 | *   VALUE(P_EXP_FILE_NAME) TYPE STRING
7 | *   VALUE(P_LANGUAGE_LIST) TYPE LOO1TAB-DATA
8 | *   VALUE(P_TRIAL_MODE) TYPE DDO2L-AS4LOCAL
9 | * EXCEPTIONS
10 | *   E_EXP_AREA_UNKNOWN
11 | *   E_EXP_AREA_NO_AUTHORITY
12 | *   E_EXP_GUI_DOWNLOADFAILED
13 | *-----
14 | *
15 | * -----
16 | * Copyright © 2001 - 2009 by Silwood Technology Ltd
17 | * -----
18 | *
19 | * Trial Mode
20 | * -----
21 | *
22 | DATA: BEGIN OF DDEEMPTY OCCURS 1,
23 |         VERSION(10) TYPE C,
24 |         END OF DDEEMPTY.
25 |
26 | IF P_TRIAL_MODE = 'Y' OR P_TRIAL_MODE = 'y' OR
27 |     P_TRIAL_MODE = 'V' OR P_TRIAL_MODE = 'v'.

```

Source Code tab showing Saphir ABAP function

Another option for loading the source code is to use the ‘Upload’ option from the ‘Utilities’ menu to load directly from the file.

Now click the ‘Attributes’ tab and click the ‘Remote-enabled module’ radio button.

Classification	
Function group	Z_SAPHIR RFC_GROUP Function Group for Saphir
Short text	Function Module for Saphir

Processing type	General data
<input type="radio"/> Normal function module	Person responsible: BCUSER
<input checked="" type="radio"/> Remote-enabled module	Last changed by: BCUSER
<input type="radio"/> Update module	Changed on: 14.07.2004
<input checked="" type="radio"/> Start immed.	Development class: Z_SAPHIR_RFCS
<input type="radio"/> Immediate start, no restart	Program name: SAPLZ_SAPHIR RFC_GR...
<input type="radio"/> Start delayed	INCLUDE name: LZ_SAPHIR RFC_GROUP ...
<input type="radio"/> Coll.run	Original language: EN
	Not released
	<input type="checkbox"/> Edit lock
	<input type="checkbox"/> Global

Function Module Attributes tab

The next step is to enter the Import Parameters.

Click the 'Import Parameters' tab and entering the parameters as shown in the form below

Attributes Import Export Changing Tables Exceptions Source code							
Parameter Name	Typing	Associated Type	Default value	Opti	Pas	Short text	Lon...
P_EXP_AREA	TYPE	DD02L - TABNAME		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Table Name	
P_EXP_FILE_NAME	TYPE	STRING		<input type="checkbox"/>	<input checked="" type="checkbox"/>		
P_LANGUAGE_LIST	TYPE	L001TAB-DATA		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Field of Length 128 for Language Impo	
P_TRIAL_MODE	TYPE	DD02L - AS4LOCAL		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Activation Status of a Repository Object	
P_EXP_TARGET	TYPE	DD02L - AS4LOCAL	'G'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Activation Status of a Repository Object	
				<input type="checkbox"/>	<input type="checkbox"/>		

Function Module Import Parameters

Ensure that the 'Pass Value' check box is checked for each parameter.

Now enter the 'Exception' parameters. Click the 'Exceptions' tab and entering the parameters as shown in the form below.

Attributes Import Export Changing Tables Exceptions Source code		
Exception	Short text	Long txt
E_EXP_AREA_UNKNOWN		Crea...
E_EXP_GUI_DOWNLOADFAILED		Crea...
E_EXP_SERVER_DOWNLOADFAILED		Crea...

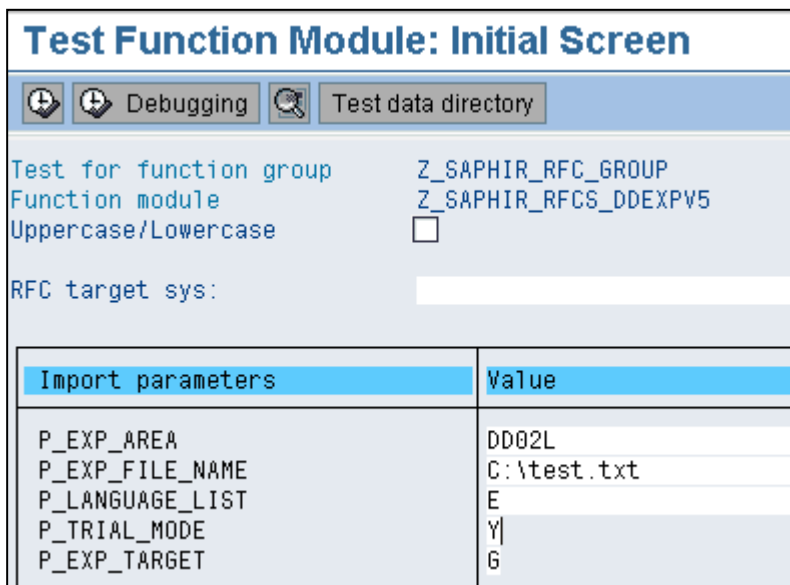
Function Module Exceptions tab

## A.4 Testing the Function

After completing all of the above steps, press CTRL+F2 to syntax check the code.

If the Check returns 'No syntax errors found', press CTRL+F3 to Activate the function.

Then click 'F8' to test the Function.



Testing the Function Module

Enter values for the Import parameters as follows:

P\_EXP\_AREA: DD02L (one of the data dictionary tables used by Saphir)

P\_EXP\_FILE\_NAME: enter a suitable location for the file created by the function

P\_LANGUAGE\_LIST: enter a valid SAP language code

P\_TRIAL\_MODE: Y (this will run the function without exporting actual data)

P\_EXP\_TARGET: G. this tests the function for execution via the SAP GUI. The file will be created on the PC where Saphir is installed. If 'S' is specified in this parameter, then the file will be created on the SAP server.

A successful result will look like the following form.

Test for function group	Z_SAPHIR_RFC_GROUP
Function module	Z_SAPHIR_RFCS_DDEXPV5
Uppercase/Lowercase	<input type="checkbox"/>
Runtime:	842.731 Microseconds
RFC target sys:	
<b>Import parameters</b>	<b>Value</b>
P_EXP_AREA	DD02L
P_EXP_FILE_NAME	C:\TEST.TXT
P_LANGUAGE_LIST	E
P_TRIAL_MODE	Y
P_EXP_TARGET	G

Result of test run on the function module

An export file will be created of zero bytes.

A further (optional) test would be to run the Function with the P\_TRIAL\_MODE set to 'N'. This will export the whole DD02L table into the named export file.

## Appendix B. The SaphirSettings.xls file

*Describes the structure and usage of the SaphirSettings.xls file*

The SaphirSettings.xls file stores most of the options used by Saphir. It is not normally necessary to modify the file manually, but there are a few circumstances where this may be necessary. These are described below.

### B.1 File usage

The SaphirSettings.xls file is used by Saphir to record a range of settings used by Saphir. The sections below describe the contents of this Excel spreadsheet, which is presented as a set of worksheets.

**IMPORTANT NOTE:** Saphir users may want to modify the spreadsheet (in particular see Appendix B of the Saphir User Guide).

Before doing this, copy the SaphirSetting.xls and use this to create a copy named SaphirSettings\_Cust.xls in the Saphir installation folder. Then add the required rules to this SaphirSettings\_Cust.xls spreadsheet.

When the Saphir application is started, if the file SaphirSettings\_Cust.xls exists, then this will be used instead of the delivered SaphirSetting.xls file.

### B.2 File structure

The file is structured as a number of worksheets. The worksheets are:

#### *DataTypeMapping*

This worksheet area shows a table of logical Saphir data types and how they map to the various RDBMS' supported by Saphir. The format is shown in the following example:

RepoDa	RepoDataType	CSAP	JDED	PSoft	SAPBW	Siebel	SiebelAly	ORAEB	S	Ot	LenFixed
S	String	CHAR;VARC	A	CHAR	CHAR;VARC	V		E;V			No
S	String	ACCP;CLNT;CUKY;LANG;UNIT			ACCP;CLNT;CUKY;LANG;U C						No
S	String			LONG							No
BN	Binary	RAW	B;O	VERS	RAW			R			No
FS	FloatSingle										8
FL	Float										12
FD	FloatDouble	FLTP			FLTP						16
N	Number	CURR;DEC;PREC;QUAN	P;S		CURR;DEC;PREC;QUAN	N		N			No
N	Number			NUM							No
N	Number			NUM							No
N	Number			SIGNED							No
N	Number			SIGNED							No
NS	NumberString	NUMC			NUMC						No

The extraction of metadata by Saphir from the ERP includes 'logical' ERP datatypes. These are specific to each ERP type. Saphir has no direct knowledge of how these logical datatypes are implemented in the physical RDBMS. This sheet presents a mapping, showing how each logical ERP datatype maps to a physical RDBMS-specific datatype.

It would not normally be necessary to amend this sheet, but if required to do so, we recommend that you consult your Saphir representative before making any changes.

*ERPTypaNaming*

This worksheet records the names for the various Enterprise Applications addressed by Saphir.

ERPTypID	ShortDesc	LongDesc
Unknown	N_D	Not Defined
SAP	SAP	SAP
JDED	JDE EntOne	JD Edwards Enterprise One
PSoft	PS Ent	PeopleSoft Enterprise
SAPBW	SAP BW	SAP BW
Siebel	Siebel	Siebel
SiebelALY	Siebel Analytics	Siebel Analytics
ORAEBS	Oracle EBS	Oracle E-Business Suite
Other1	Other ERP 1	Other ERP Type 1
Other2	Other ERP 2	Other ERP Type 2
Other3	Other ERP 3	Other ERP Type 3

The columns are as follows:

- ERPTypID: This is the internal identifier for the ERP type
- ShortDesc The short description that will appear in the Saphir interface
- LongDesc The long description that will appear in the Saphir interface

*ERPTypLanguages*

This worksheet lists the available languages for extraction of descriptive fields from each Enterprise Application. A typical set of language codes for each Enterprise Application are coded in this worksheet, but you may need to add your own if you wish to extract metadata in a language not included in the list. The worksheet format is shown in the following example:

RepoLanguageID	LanguageDesc	LanguageNotActive	SAP	JDED	PSoft	SAPBW	Siebel	SiebelAly	ORAEBS	Other1	Other2	Other3
E	English		E	E	ENG	E	E	E	US	E	E	E
D	German		D	D	GER	D	D	D		D	D	D
I	Italian		I	I	ITA	I						
F	French		F	F	FRA	F						
S	Spanish		S	S	ESP	S						
N	Dutch		N		DUT	N						
P	Portuguese		P			P						
V	Swedish	X	V			V						
R	Russian	X	R			R						
C	Czech	X	C			C						
1	Chinese	X	1			1						
J	Japanese	X	J			J						
3	Korean	X	3			3						

The columns are as follows:

- RepoLanguageID: This is the internal identifier for the ERP type

- LanguageDesc                      A descriptive name for the language
- LanguageNotActive                If blank, then the language is active. If 'X' then the language is not currently supported by Saphir
- SAP, JDED, etc....                The 'native' language code for that ERP that corresponds to the Repository Language code. For example, the PeopleSoft language code for English is 'ENG' – this maps to the Saphir internal language code of 'E'. So when Saphir performs the metadata extraction from a PeopleSoft system where the user has requested 'English', Saphir uses the worksheet to select the PeopleSoft-specific language code to use.

IMPORTANT NOTE: Saphir does not yet support non-western character sets. Do not try to perform an extraction of metadata from your ERP using any of the language codes shown as Inactive in the image above.

*PSoftRuleBasedRelationships*

This worksheet is used by the Rules Based Relationship Generation feature of Saphir to create additional relationships for a PeopleSoft Repository. Please see the Saphir User Guide for more details on how to use this feature. The worksheet format is shown in the following example:

A	B	C	D	E	
ChildTableName	ChildFieldNames	ParentTableName	RuleNotActive	ExtRelGenParent	Remark
AAP_YEAR_GOALS		AAP_YEAR			
AAP_YEAR_JOBGRP		AAP_YEAR			
AAP_YEAR_JG_GLS		AAP_YEAR_JOBGRP			
VENDOR_ADDR		VENDOR		X	VENDOR shall be used
VENDOR_ADDR_LNG	+EFFDT	VENDOR_ADDR	X		just and example to sh
VENDOR_ADDR_PHN		VENDOR_ADDR			
VENDOR_ADDR_TMP		VENDOR_ADDR			
VENDOR_CNTCT		VENDOR			
VENDOR_LOC		VENDOR			
VENDOR_LOC_TMP		VENDOR_LOC			
VENDOR_PAY		VENDOR_LOC			
VENDOR_PAY_TMP		VENDOR_PAY			
VENDOR_POLICY		VENDOR			
VENDOR_SF_TBL	SETID;VENDOR_ID;	VENDOR			
VENDOR_SF_TBL	EXT_ORG_ID;*	VENDOR			
EPPCM_CATG_HIER		EPPCM_CATEGORY			default mapping
EPPCM_CATG_HIER	EPPCM_PARPORTALNM;EPPCM_PARCATGID;	EPPCM_CATEGORY			rolenamed mapping

The columns are as follows:

- ChildTableName:                    The name of the Child table for the relationship.
- ChildFieldNames:                See 'Defining a Rule' in Appendix B of the Saphir User Guide for details of the possible values.
- ParentTableName:                The name of the Parent table for the relationship.
- RuleNotActive:                  If blank, then the rule is active. If 'X' then the rule will be ignored.
- ExtGenRelParent                Used by the Extended Relationship generation process (See Appendix B of the Saphir user Guide for details).
- Remark:                          A free-format comment area for entry of optional notes describing the rule

*JDERuleBasedRelationships*

This worksheet has the same format as the PSoftRuleBaseRelationships worksheet. It is used by the RulesBased Relationship Generation feature of Saphir to create additional relationships for a JDEdwards Repository. See Appendix B of the Saphir user Guide for more details.

ChildTableName	ChildFieldNames	ParentTableName	RuleNotActive	ExtRelGenParent	Remark
F0015		F0013		X	Default Parent for
F0015	+REPLLAST;T1CRDC	F0013			
F08320	BAANN8	F0101			
F083202X		F083202			
F08320B		F083202			
F08320C		F083202			
F08320WF		F083202			
F08320X		F08320			
F08330		F083202			
F083303		F083302			
F083303		F069016			
F083304		F083302			
F083304		F083301			
F083305		F083301			
F08330B		F083202			
F08330B	+REPLLAST;BTAOPT	F083202			
F08330C		F083202			
F08330C	+REPLLAST;PCAOPT	F083202			
F08330WF		F083202			
F08330WF	+REPLLAST;BRAOPT	F083202			

## Appendix C. Using SAPRFC.ini with Saphir

*Describes how to use the SAPRFC.ini file to attach to SAP systems with Saphir*

The Saphir SAP RFC connection method uses a set of parameters to establish connection with a SAP instance. These are described in the section ‘Entering the SAP RFC connection settings’ in Chapter 2 of this manual. The SAPRFC.ini file allows a richer set of parameters to be used where the standard set shown in the Saphir are not sufficient to perform a satisfactory connection.

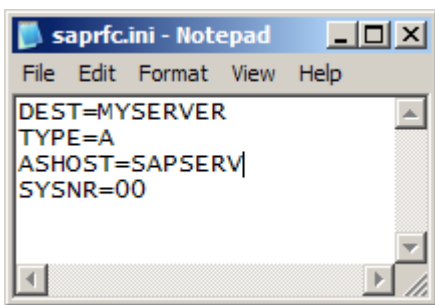
### C.1 Creating the SAPRFC.ini file

The SAPRFC.ini file is referenced by components of the SAPgui software which must be installed on the PC where Saphir is installed.

If there are other applications accessing SAP from the PC, then SAPRFC.ini may already exist. It may be located in the following folder:

`\Program Files\SAP\FrontEnd\SAPgui\rfcsdk\`

If the file does not exist, use Notepad or a similar text editor to create the file structure as shown below.



Structure of the saprfc.ini file

The connection to the SAP system can now be tested using the ‘sapinfo.exe’ program. Use a command line session as shown in the following screen shot.

```
C:\Program Files\SAP\FrontEnd\SAPgui\rfcsdk\bin>sapinfo.exe DEST=MYSERVER

SAP System Information
-----

Destination          susap1_GS0_00
Host                 susap1
System ID            GS0
Database             GS0
DB host              SUSAP1
DB system            ORACLE

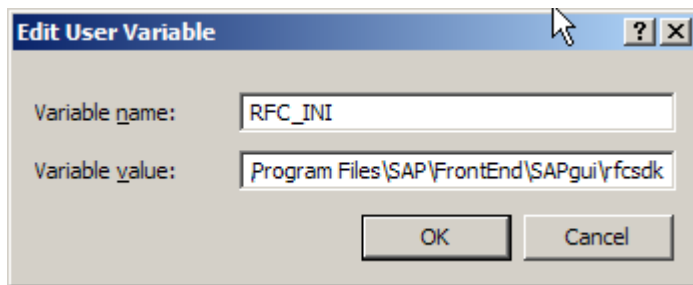
SAP release          620
SAP kernel release   620

RFC Protokoll        011
Characters           1100 <NON UNICODE PCS=1>
Integers             LII
Floating P.          IE3
SAP machine id       560

Timezone              3600 <Daylight saving time>
```

using 'sapinfo' to check connection to the SAP system

In order to use the SAPRFC.ini from any RFCSDK application, the file needs to be registered as a system variable. From the Windows Control panel, select 'System', then on the 'Advanced' tab click 'Environment Variables' and create a new variable as shown in the next screen shot.

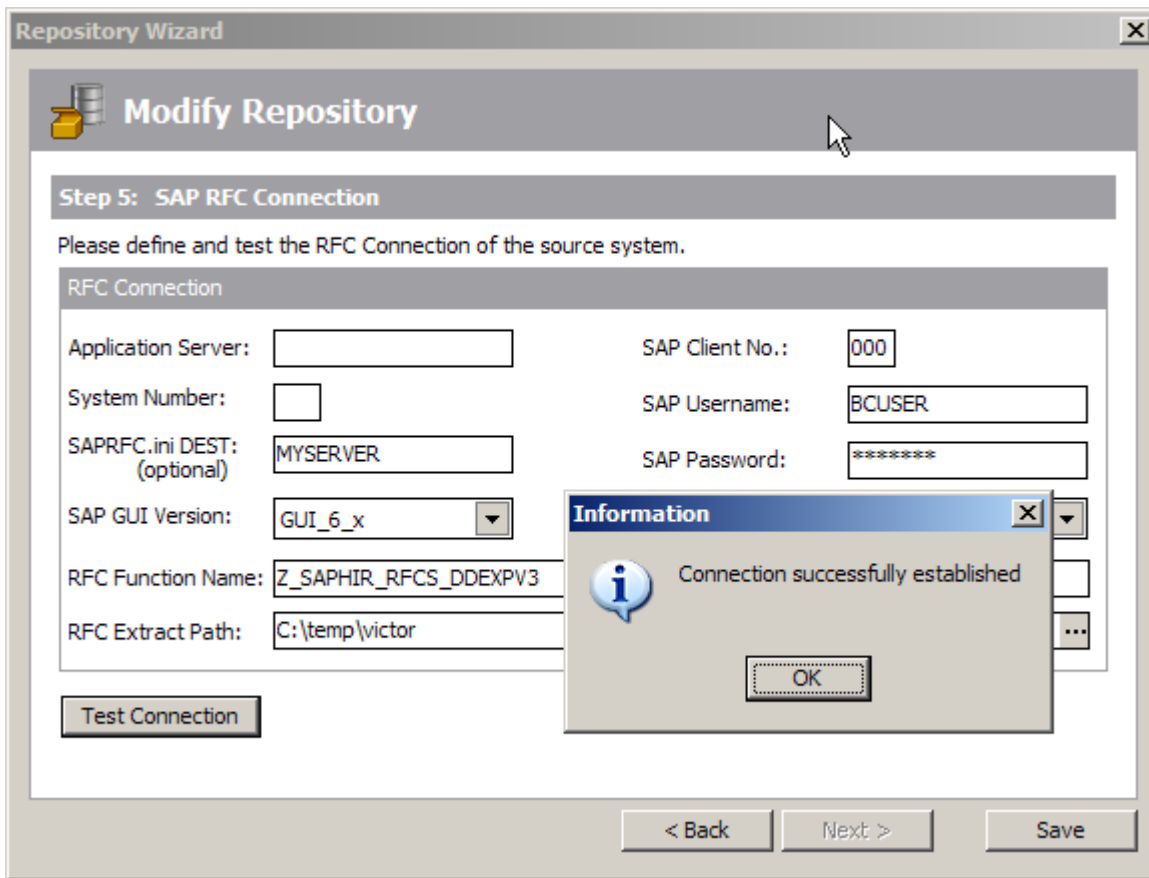


Creating the RFC\_INI Environment Variable

## C.2 Using SAPRFC.ini from Saphir

To use SAPRFC.ini with Saphir, enter the required SAP 'Destination' name into the 'SAPRFC.ini DEST' field on the Saphir RFC Connection form. There is no need to enter an Application Server or System Number. Clicking the 'Text Connection' button should then establish a connection with the appropriate SAP system (see screen shot below).

For full details of how to use SAPRFC.ini see the SAP 'RFCSDK Guide'. This manual describes all the possible parameters for inclusion in the SAPRFC.ini file.



Using the SAPRFC.ini Destination to connect to SAP

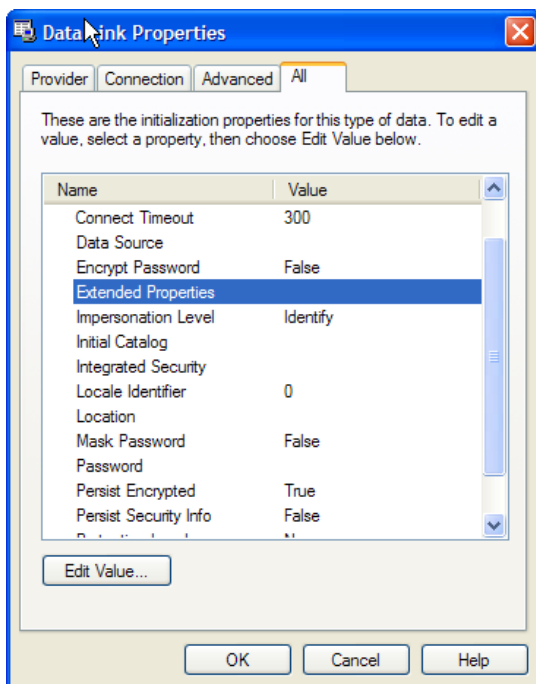
## Appendix D. Connecting to a SQLAnywhere database

*Describes how to configure an ADO connection to a SQLAnywhere database*

It is important to use an ADO connection to a SQLAnywhere database, rather than using ODBC. This will lead to faster extraction times from the ERP system. Chapter 2 describes how to perform the first few steps in configuring such a connection. The additional settings are described below.

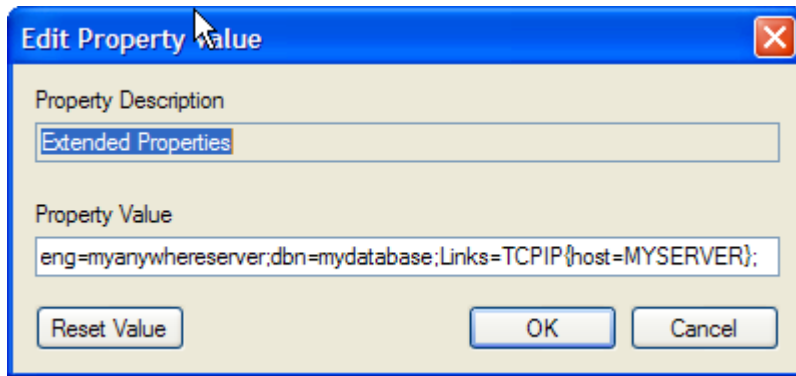
### D.1 Settings for connecting to the database

Having selected SQL Anywhere OLDEB Provider on the 'Provider' tab of the Data Link Properties screen, click on the 'All' tab to display the screen as shown below.



'All' tab on the Data Link Properties screen

Click on the 'Extended Properties' and then select the 'Edit value' button.



Editing Extended Properties

Now edit the 'Property Value' as shown above.

The string is of the form: `eng=<dbserver>;dbn=<dbname>;Links=TCPIP{host=<servername>};`

Where the variables are as follows:

dbserver:      The name of the SQLAnywhere Server on which the database resides

dbname:        The name of the SQLAnywhere database

servername:    The name of the hardware server on which the database resides.

After entering these values, click the OK button.

Now click the 'Connection' tab and verify the connection is working by clicking the 'Test Connection' button.

## Appendix E. Using the ABAP 'ALL' Functions

*Using a second ABAP Function to call the Main Saphir ABAP Function without using SAP RFCs*

Appendix A describes how to install the main ABAP Function supplied with Saphir. This will be based on one of two source ABAP files supplied with Saphir, one being for SAP and the other for SAP BW. Saphir invokes this ABAP function using SAP RFC calls. Saphir is supplied with 2 additional ABAP files that can be used when invocation by RFC is either not possible or not permitted by the SAP team.

Both are very simple functions that call the main ABAP function to extract metadata from the SAP or SAP BW system. Both can be run in two different ways:

- 1) To create the files on the 'local' PC. The generated ABAP files will be directly available to Saphir so they can be processed by Saphir to populate the Saphir repository. This method uses a SAP feature called GUI\_DOWNLOAD. GUI\_DOWNLOAD moves files from the server onto a local PC and this cannot be run in 'batch' mode.
- 2) To create the files on the SAP server. The generated ABAP files will be created on a local SAP Server drive and will need to be copied to a drive visible to Saphir before they can be processed by Saphir to populate the Saphir repository. This method does not use GUI\_DOWNLOAD and can therefore be run in 'batch' mode. It is the responsibility of the Saphir user to create a suitable batch program for scheduling the ABAP function when it has been decided to run in batch mode.

### E.1 Installing the 'All' Functions

Installation of these functions is very similar to installing the main function, as described in Appendix A.

The Function should be installed in the same Function Group as the 'main' function.

Open the appropriate delivered ABAP text file with Notepad and copy the full text of the file to the clipboard.

These files are installed in the ABAPFiles folder below the default Saphir folder and are as follows:

- For a SAP system: z\_saphir\_rfcs\_ddexpv5\_all.txt.
- For a SAP BW system: z\_saphir\_rfcs\_bwexpv5\_all.txt.

From the Object Navigator, right mouse click on the Function Group previously created for the main ABAP function (See Appendix A) and click 'Create' and then choose 'Function Module'.

On the 'Create Function Module' form, enter a name and a Short text description for the Module and click the Save icon. The name entered should reflect that of the chosen ABAP Function. That is:

For a SAP system: Z\_SAPHIR\_RFCS\_DDEXPV5\_ALL

For a SAP BW system: Z\_SAPHIR\_RFCS\_BWEXPV5\_ALL

The Function Module editor now appears. Click on the 'Source code' tab and replace the contents of the workspace with the contents of the clipboard copied from the appropriate ABAP 'all' function earlier.

Another option for loading the source code is to use the 'Upload' option from the 'Utilities' menu to load directly from the file.

Now click the 'Attributes' tab and click the 'Remote-enabled module' radio button.

The next step is to enter the Import Parameters.

Click the 'Import Parameters' tab and entering the parameters as shown in the form below

Parameter Name	Typing	Associated Type	Default value	Opti	Pas	Short text	Lon...
P_EXP_FILE_PATH	TYPE	STRING		<input type="checkbox"/>	<input checked="" type="checkbox"/>		...
P_LANGUAGE_LIST	TYPE	L001TAB-DATA		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Field of Length 128 for Language Impo	...
P_TRIAL_MODE	TYPE	DD02L-AS4LOCAL		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Activation Status of a Repository Object	...
P_EXP_TARGET	TYPE	DD02L-AS4LOCAL	'G'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Activation Status of a Repository Object	...
				<input type="checkbox"/>	<input type="checkbox"/>		...

Function Module Import Parameters

Ensure that the 'Pass Value' check box is checked for each parameter.

## E.2 Testing the Function

After completing all of the above steps, press CTRL+F2 to syntax check the code.

If the Check returns 'No syntax errors found', press CTRL+F3 to Activate the function.

Then click 'F8' to test the Function.

Import parameters	Value
P_EXP_FILE_PATH	
P_LANGUAGE_LIST	
P_TRIAL_MODE	
P_EXP_TARGET	G

Testing the Function Module

Enter values for the Import parameters as follows:

P\_EXP\_FILE\_PATH: enter a suitable location for the files created by the function

P\_LANGUAGE\_LIST: Enter a valid SAP language code

P\_TRIAL\_MODE: Y (this will run the function without exporting actual data)

## USING THE ABAP 'ALL' FUNCTIONS

P\_EXP\_TARGET: Set this to 'G' to create files via GUIDOWNLOAD onto a PC, or 'S' to create the files on a SAP Server (using Dataset transfer).

An export file will be created of zero bytes.

# Index

- ABAP
  - Extracting Metadata with, 18
- Application Hierarchy, 75
- Database user, 8
- Databases
  - communication with, 7
- DB2, 8
- EnterpriseOne
  - XML files, 31
- Exporting to third party tools, 75
- Extracting Metadata
  - restarting, 68
- J.D.Edwards EnterpriseOne
  - Extracting Metadata, 61
  - Versions supported, 4
- Metadata
  - browsing, 69
- Oracle, 8
- Oracle EBS
  - Extracting Metadata, 65
- PeopleSoft Enterprise
  - Extracting Metadata, 54
  - Versions supported, 4
- Repository Maintenance, 17
- Repository Manager, 9
- SAP
  - ABAP Connection Method, 18
  - ABAP Function, 77
  - BW Versions supported, 4
  - Connecting, 18
  - Extracting Metadata, 48
  - RFC Connection Method, 25
  - Versions supported, 4
- SAP BW, 77
- SAP GUI, 6
- Saphir
  - Installation, 8
  - Manuals, 4
  - Setup, 6
- Saphir database
  - creating, 7
- Saphir Repository
  - Opening, 47
- SaphirSettings.xls file, 84
- SAPRFC.ini, 88
- Siebel
  - Extracting Metadata, 58
  - Versions supported, 4
- SQL Anywhere, 8
  - connecting, 91
- SQL Server, 8
- SQLite, 8
- Starting Saphir, 8
- Subject Areas, 73
- System and resource requirements, 6