

White Paper

Configuration of Relational Reporting

Fabasoft Folio 2019

Copyright © Fabasoft R&D GmbH, Linz, Austria, 2019.

All rights reserved. All hardware and software names used are registered trade names and/or registered trademarks of the respective manufacturers.

No rights to our software or our professional services, or results of our professional services, or other protected rights can be based on the handing over and presentation of these documents.

Contents

1 Introduction	4
2 Software Requirements	4
3 Prerequisites	4
3.1 Database	4
3.2 Fabasoft Folio Domain	4
3.3 Fabasoft Folio AT Service	6
4 Creating a Relational Report Definition	7
4.1 Defining the Table Columns	8
4.1.1 Defining the Table Columns Based on a View	8
4.1.2 Defining the Table Columns Manually	8
5 Creating a Relational Report	9
5.1 Creating a Relational Report Manually	10
5.1.1 Scheduling	10
5.1.2 SQL Processing and Execution Information	11
5.2 Creating a Relational Report Using the “Transfer to Report” Feature	11
6 Example: Use of LibreOffice Base as Reporting Tool	12
6.1 Microsoft Windows Environment	12
6.1.1 Install PostgreSQL ODBC Driver	12
6.1.2 Connect from LibreOffice Base to the Database	14
6.2 Linux Environment	16
6.2.1 Define the Java Runtime Environment Used by LibreOffice	16
6.2.2 Connect from LibreOffice Base to the Database	17

1 Introduction

The software component *Relational Reporting* (FSCRELREP) allows saving data from Fabasoft Folio in a relational database (Microsoft SQL Server, Oracle or PostgreSQL). This data can be processed with an external reporting tool.

2 Software Requirements

System environment: All information contained in this document implicitly assumes a Microsoft Windows environment or Linux environment.

Supported platforms: For detailed information on supported operating systems and software see the software product information on the Fabasoft distribution media.

Descriptions in this document are based on the following software:

Database:

- Microsoft SQL Server 2016 Enterprise SP2 (x64)
- Oracle Database 12c Release 2 Enterprise Edition (from 12.2.0.1.0) for Linux x86-64 with RAC
- PostgreSQL 11.2

Client workstation (only when using LibreOffice Base):

- LibreOffice Base 6.1.5

3 Prerequisites

3.1 Database

Create a database for storing the exported data. During the definition of the database resource in the Fabasoft Folio Domain, the connection settings of this database have to be specified in the corresponding property fields.

3.2 Fabasoft Folio Domain

The following steps have to be performed in the Fabasoft Folio Domain:

- Verify that the software component *Relational Reporting* (FSCRELREP) is installed in the Fabasoft Folio Domain.
- In the *Current Domain* on the "System Configuration" tab in the *Default Datasource (Reporting)* (FSCRELREP@1.1001:defaultdbresource) field create a *Datasource* (FSCRELREP@1.001:DBresource) with the correct connection data to the database created in chapter 3.1 "Database".

Microsoft SQL Server connection:

Reporting (SQL Server) (Datasource): Edit

Cancel Apply Next

Datasource Component Object Object Versions Security Signatures

Multilingual Name One Entry

Language *	Language-Specific String *
1 English	Reporting (SQL Server)

Connection Data *

Connection Type *	SQL Server
Data Source	dbsql.testcenter.com
Database	Reporting
Port (Only PostgreSQL)	
Database Login Name	
Database Login Password	
Enable Tracing	<input type="checkbox"/>

PostgreSQL connection:

Reporting (PostgreSQL) (Datasource): Edit

Cancel Apply Next

Datasource Component Object Object Versions Security Signatures

Multilingual Name One Entry

Language *	Language-Specific String *
1 English	Reporting (PostgreSQL)

Connection Data *

Connection Type *	PostgreSQL
Data Source	dbpgs.testcenter.com
Database	Reporting
Port (Only PostgreSQL)	5432
Database Login Name	repuser
Database Login Password	●●●●●●●●
Enable Tracing	<input type="checkbox"/> Undefined

Oracle Database connection:

Reporting (Oracle) (Datasource): Edit

Buttons: Cancel, Apply, Next

Navigation: Datasource, Component Object, Object, Versions, Security, Signatures

Multilingual Name (One Entry)

Language *	Language-Specific String *
1 English	Reporting (Oracle)

Connection Data *

Connection Type *	Oracle
Data Source	orcl
Database	
Port (Only PostgreSQL)	
Database Login Name	repuser
Database Login Password
Enable Tracing	<input type="checkbox"/> Undefined

3.3 Fabasoft Folio AT Service

To create and configure a Fabasoft Folio AT Service, perform the following steps:

1. Install *Fabasoft Folio AT Services*. Open the Fabasoft Folio Server Management and navigate to the node "Fabasoft Folio/AT". Create and start a new Fabasoft Folio AT Service, if there is none.
2. Make sure that the Fabasoft Folio AT Service user has administrative rights on the database, because schemas are created in the context of this user.
3. Additionally the user has to be a privileged user in the Fabasoft Folio Domain and needs the right to search relational reports and to read associated report definitions.
4. In the Fabasoft Folio Domain create an *Automated Task* (Domain Administration > User Objects > Automated Tasks) and assign this task to the Fabasoft Folio AT Service user. The *Automated Task* has to run the *Check for waiting relational report jobs* action (FSCRELREP@1.1001:CheckForReportJobs). In a defined interval, this action searches for relational reports, which show a past related entry in the *New execution from* field and initiates the calculation. The *New execution from* field is set to the current date if the report is opened with a right-click. The date can be modified on the "Scheduling" tab in the *Start Execution* field. **Note:** In order to kill stalled automated tasks, the Fabasoft Folio AT Service is restarted automatically after a defined period of time. The value for restarting the Fabasoft Folio AT Service is defined in the *Restart after (min)* field of the Fabasoft Folio AT Service (or the *Automated Tasks*). The default value is 60 minutes. It is important to adapt the value so that all tasks – this means the calculation of all open relational reports – can be processed within this

interval (e.g. 200 minutes).

Check for Relational Report Jobs (AT) (List of Automated Tasks): Edit

Versions Security Digital Signature Workflow Additi

List of Automated Tasks Signatures Object With Object List (Advanced)

Tasks

Name	Start on/at	State
Automated Task (Reporting)		Activated
Check list of background activities		Activated

Maximum Number of Threads	4
Maximum Run Time (Min.)	
Wait for Lock (Sec.)	10
Restart After (Minutes)	200

Resources

Resource Maximum Run Time (Min.) Maximum Number of Threads Restart After (Minutes)

4 Creating a Relational Report Definition

A relational report definition is represented by an instance of object class *Relational Report Definition* or object class *Relational Report Definition (Component Object)*.

Each user can define a relational report based on a relational report definition. For the relational report of a user, only objects according to the user's access rights will be included.

In the properties of this object several settings can be defined:

- *Report Reference*
This property defines a reference string, which will be used to prefix the names of the tables generated in the database.
- *Table Definition*
This compound property allows you to define the properties of the tables to be generated by a relational report.

Note: For a relational report, at least one table is generated in the database. Depending on the column definitions, additional tables might be created for compound property lists (such as the list of addresses of a person) and object paths (such as the list of contact persons of an organization).

The following properties can be defined in the *Table Definition* compound property:

- *Table Reference*
This property is used for generating the name of the main table of the relational report. The table name is prefixed with the string defined in the *Report Reference* property and an underscore (e.g. "ReportReference_TableReference").
- *Selection*
This property allows you to specify a predefined selection object that is used for populating the *Selection* property of relational reports based on this relational report definition.
- *Columns*
This property defines the table's columns. The columns of the table can be defined manually or automatically based on a view. The two options are described in the following chapter.
- *Allowed Object Classes*
This property lets you restrict the permitted object classes for a relational report definition. If object classes are specified in the *Allowed Object Classes* property of a relational report definition, only instances of these object classes are exported to the database when a relational report referencing the relational report definition is executed. Instances of other object classes are not included in the relational report. Additionally, if allowed object classes are defined for a relational report definition, the *Included Objects* property will not allow you to add instances of object classes other than the ones specified in the *Allowed Object Classes* property.
- *Access Types*
This property allows you to specify an optional list of access types to be checked for every object included in a relational report. If the ACL of an object to be exported to the database does not grant the specified access types to the user executing the relational report, the object is not exported to the database. Instead, the object is ignored and processing is continued with the next object to be exported.

4.1 Defining the Table Columns

4.1.1 Defining the Table Columns Based on a View

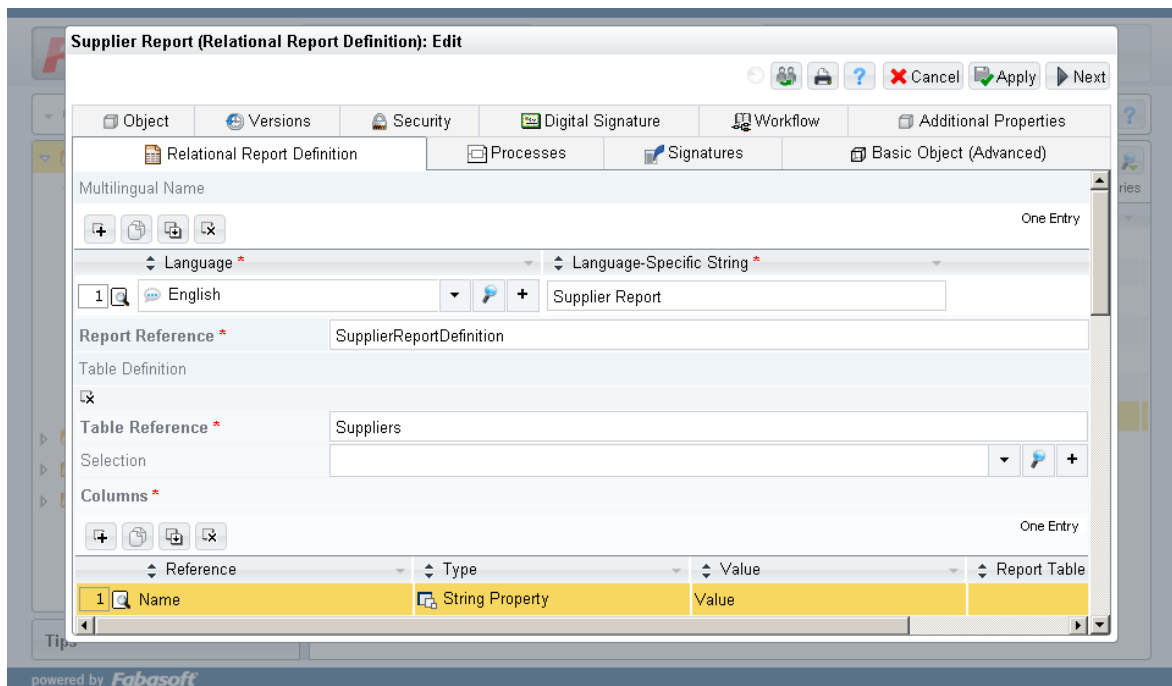
The easiest and most convenient way to set up a relational report definition is to define the columns based on a view.

Therefore you can simply copy the column settings from an object list and paste it onto the relational report by clicking "Paste Column Settings" from the relational report's context menu. The columns are automatically created in the *Columns* property.

Note: You can also manually edit the columns defined in a relational report definition after populating it based on the column settings of an object list.

4.1.2 Defining the Table Columns Manually

In the *Columns* property of a relational report definition you can manually define the columns of the main table to be exported to the database.



You can define multiple columns as follows:

- *Reference*
In this property, you have to provide a name for the column. The column name must be compliant to the SQL standard.
- *Type*
This property specifies the data type of the column.
- *Value*
This property determines how to retrieve or calculate the values stored in this column. It allows you to reference a property path in the *Follow This Property Path* property, a property in the *Use Values From This Property* property or an expression in the *Or Else Calculate This Expression*. If you provide a property or property path, the expression is automatically generated from the selected values. For each object to be exported by a relational report, each column expression is evaluated in order to determine the value to be stored in the database.
- *Report Table*
This property allows you to specify a table definition represented by an instance of object class *Report Table* or object class *Report Table (Component Object)*. This way, you can define additional tables for your relational report definition and resolve compound properties, object pointer properties, and object lists so they are exported to separate tables in the database.

5 Creating a Relational Report

A relational report is represented by an instance of object class *Relational Report*.

Relational reports can either be created manually or using the *Transfer to Report* feature.

5.1 Creating a Relational Report Manually

Each user can only create one relational report for each relational report definition. If a relational report already exists for the selected relational report definition, Fabasoft Folio will ask you to edit the existing relational report instead.

In the properties of a *Relational Report* several settings can be defined:

- *Database Resource*

In this property, select the database you want to connect to. If you are using Microsoft SQL Server, you can use the *Default Database (SQL Server)* object shipped with Fabasoft Folio. Otherwise, you may need to create a new *Database Resource* object where you can specify the connection settings needed to connect to your database.

Note: In the *Default Database Resource (Reporting)* property on the *System Configuration* form page of the current domain, you can reference a default database resource, which is then used to populate the *Database Resource* property of relational reports when they are created.

- *State*

This property shows the current state of the database. After a relational report has been executed successfully, its state is changed from "Empty" to "Valid". In addition to this, the *Generated on/at* property is populated with the date and time of the last export.

- *Active*

To temporarily prevent a relational report from being processed, set this property to "No".

- *Selection*

In this property, you can reference instances of the object classes *Selection Result*, *Selection Definition*, *Selection Definition (Component Object)*, and *Static Object List* which provide objects to be included in the relational report.

- *Included Objects*

The *Included Objects* property allows you to reference a static list of objects to be included in the report.

Note: The *Allowed Object Classes* and *Access Types* properties of the relational report definition associated with the relational report can be used to filter the set of objects that is eventually exported to the database when the relational report is processed.

5.1.1 Scheduling

A relational report is not executed unless scheduled for execution by the Fabasoft Folio AT Service.

To schedule a relational report for execution, click "Enquire Report" from the context menu of the relational object. This will set the *Start Execution on/at* property to the current date and time.

You can also manually set this property to the desired value.

Note: The relational report is processed when the Fabasoft Folio AT Service executes the automated task *Check for waiting relational report jobs* for the next time on or after the date and time specified in the *Start Execution on/at* property. The relational report is not processed before the specified date and time.

On the *Scheduling* form page you can also find the following properties:

- *Repeat Execution Every*

This property allows you to schedule a relational report for recurring execution. This feature, for example, allows you to update the data in your database daily or weekly.

- *Delete Report After*
Additionally, you can define the *Delete Report after* and *Repetition Type* properties to have the *Check for waiting relational report jobs* task periodically delete the exported data from the database.
- *History (Last 10)*
This property shows a log of the ten most recent executions of the relational report.

5.1.2 SQL Processing and Execution Information

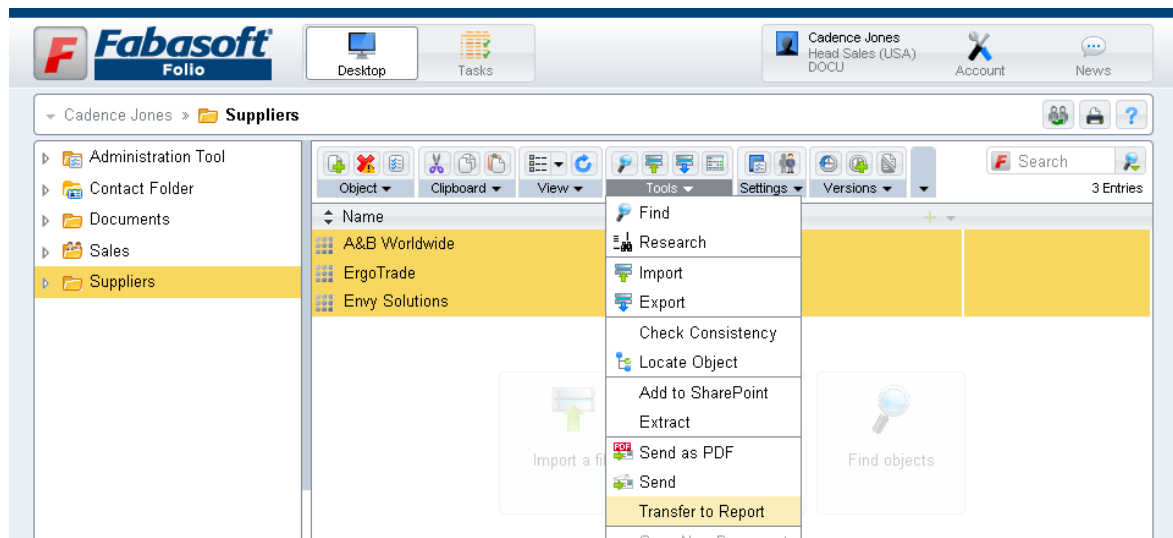
In the *Process SQL* property on the *Advanced* form page, you can define SQL code that is executed after the data for your relational report has been exported to the database by the automated task *Check for waiting relational report jobs*.

Note: You can also define SQL code in the *Process SQL* property of a relational report definition. This SQL code is then used to populate the *Process SQL* property when a new relational report is created based on the relational report definition.

After the execution of the relational report, a status message is stored in the *SQL Execution Information* property informing you whether the optional SQL code defined in the *Process SQL* property had been executed successfully or not.

5.2 Creating a Relational Report Using the “Transfer to Report” Feature

Fabasoft Folio allows you to create a relational report based on one or more objects selected in an object list.



Note: A “Transfer to Report” button is also available in the search results dialog box so you can select one or more objects in a search result list and create a relational report based on these objects.

Note: For this feature to work, you need to make one or more relational report definitions available by adding them to a template collection. In addition to this, you must be granted the *Read Properties* access type by the ACL of the relational report definition to be able to use it as a template for new relational reports.

If there is only one matching relational report definition available in your template collections, a relational report is created based on this relational report definition. If there are more than one

matching relational report definitions available in your template collections, you have to select the relational report definition you want to use for your new relational report. Otherwise, if no usable relational report definitions are available in your template collections, an error message is displayed.

A relational report definition referenced in a template collection is only included in the list of available relational report definitions if the selected objects are instances of the *Allowed Object Classes* referenced by the relational report definition. If no object classes are specified in the *Allowed Object Classes* property of a relational report definition, it can be used with instances of all object classes.

After selecting the relational report definition, a relational report is created and opened to allow you to edit its properties.

The *Included Objects* property is populated with the objects you selected before clicking "Transfer to Report". Moreover, the *Start Execution on/at* property is initialized with the current date and time to automatically schedule the relational report for immediate execution.

The remaining properties can be defined in the same way as you would define them when manually creating a relational report.

6 Example: Use of LibreOffice Base as Reporting Tool

LibreOffice Base connected to a PostgreSQL database can be used as reporting tool. This section describes how to configure the connection.

6.1 Microsoft Windows Environment

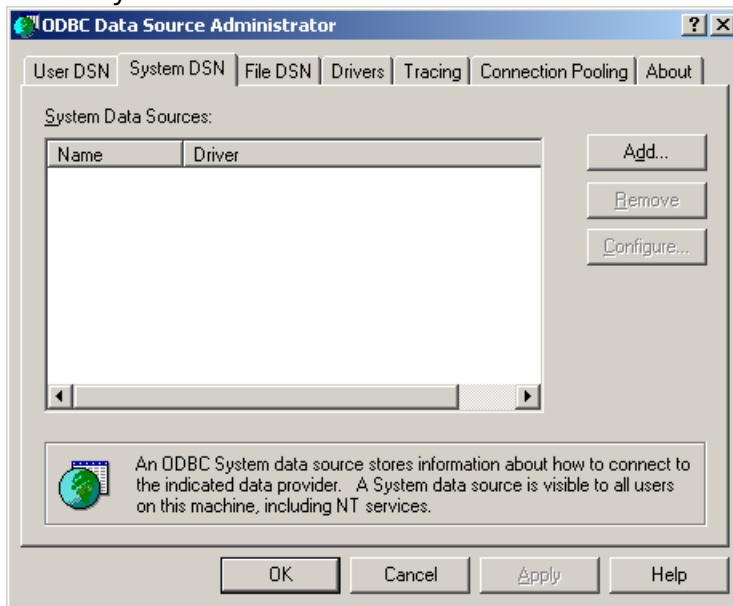
To connect from LibreOffice Base to a PostgreSQL database the PostgreSQL ODBC driver is required. This driver can be downloaded from <http://www.postgresql.org/ftp/odbc/versions/dll/>.

6.1.1 Install PostgreSQL ODBC Driver

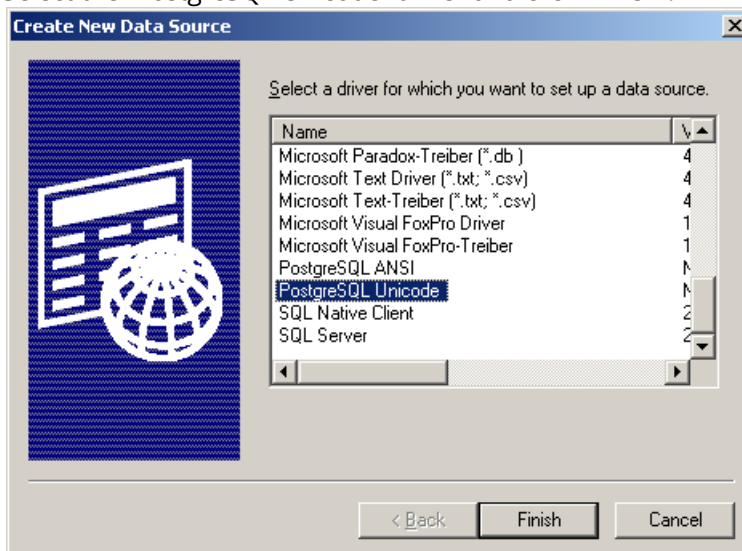
To install the PostgreSQL ODBC driver, perform the following steps:

1. Click "Start" > "Run" and enter `odbcad32` to start the ODBC data source administration tool.
2. Click "OK".

3. On the "System-DSN" tab click "Add".



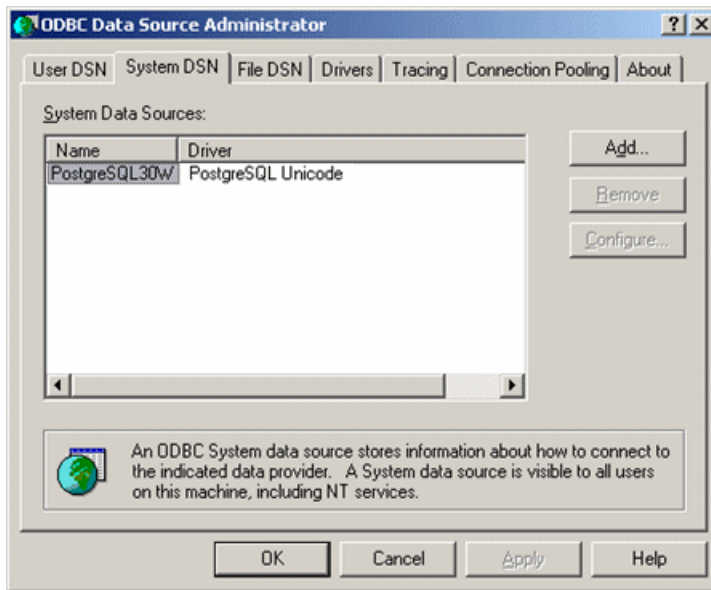
4. Select the "PostgreSQL Unicode" driver and click "Finish".



5. Define the database connection.



6. Click "OK".



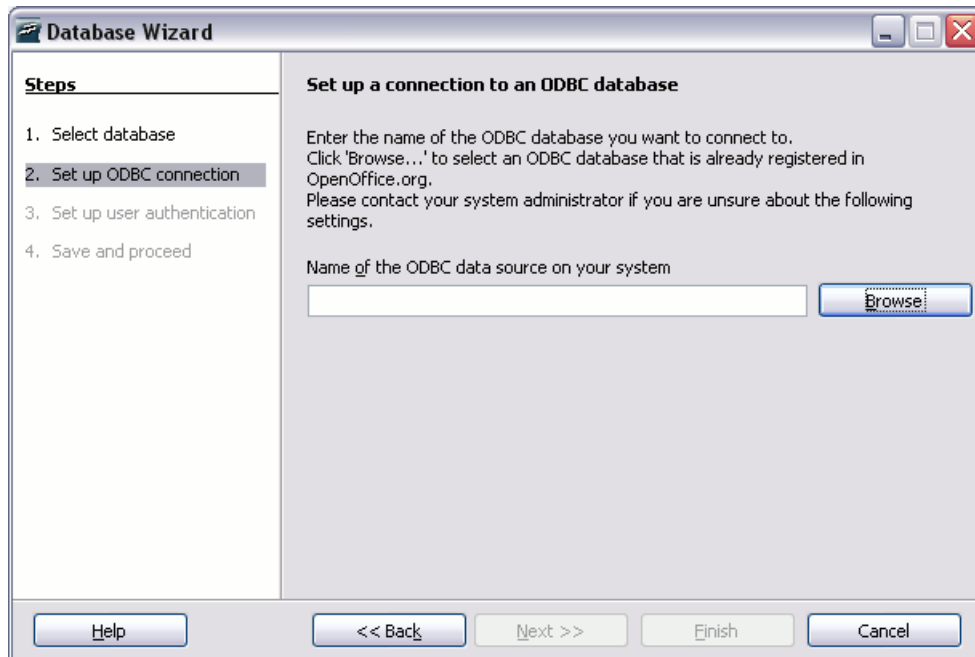
6.1.2 Connect from LibreOffice Base to the Database

To connect from LibreOffice to a PostgreSQL database, perform the following steps:

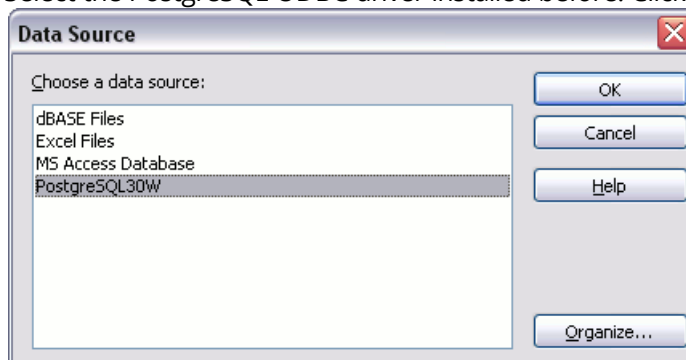
1. Start LibreOffice Base.
2. In the LibreOffice Database Wizard select "Connect to an existing database" and "ODBC". Click "Next".



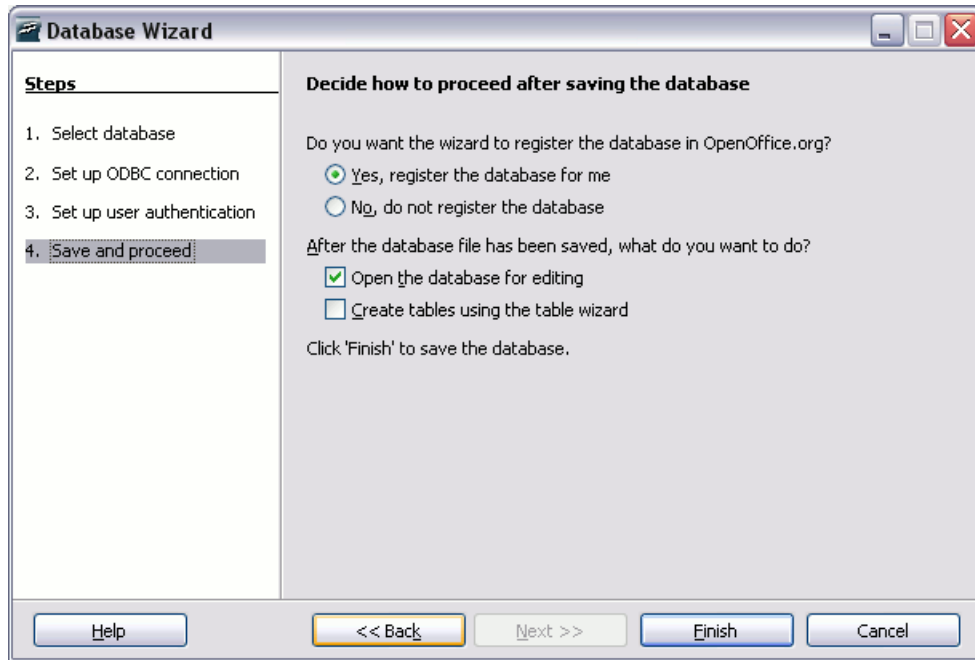
3. Define the ODBC connection. Therefore click "Browse".



4. Select the PostgreSQL ODBC driver installed before. Click "OK".



5. Click "Next".
6. Select "Yes, register the database for me" and "Open the database for editing". Afterwards click "Finish".



LibreOffice Base is started and queries and reports can be created.

6.2 Linux Environment

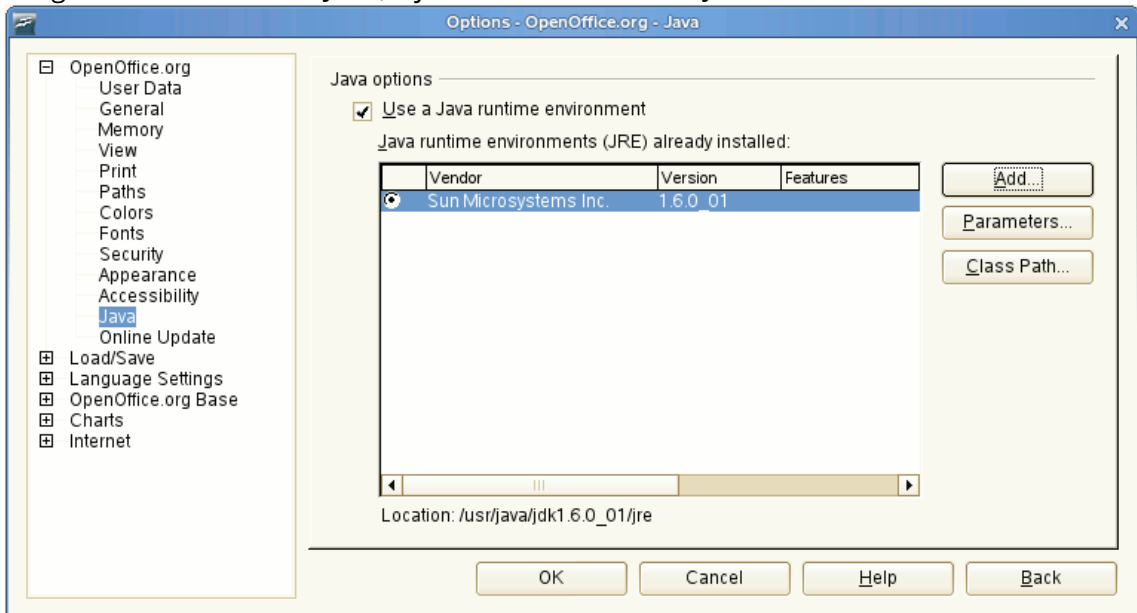
In a Linux environment the connection from LibreOffice Base to PostgreSQL should be realized with JDBC. The PostgreSQL JDBC driver can be downloaded from <http://jdbc.postgresql.org/download.html>. The download version depends on the version of Java Runtime Environment (JRE) that is used by LibreOffice and on the version of PostgreSQL.

6.2.1 Define the Java Runtime Environment Used by LibreOffice

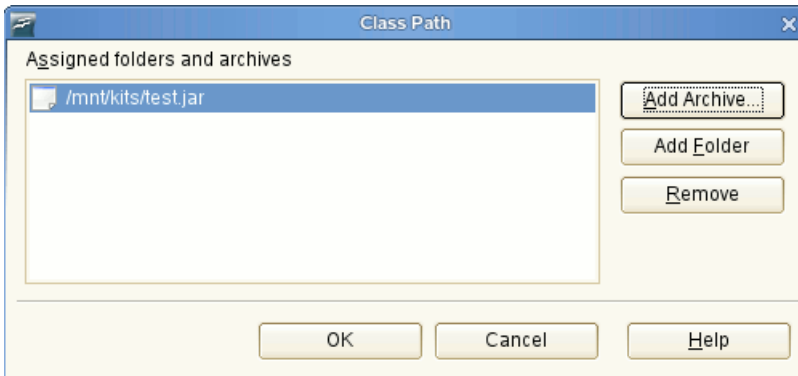
To define the Java Runtime Environment used by LibreOffice, perform following steps:

1. Start the LibreOffice Writer.
2. Select "Tools" > "Options".

3. Navigate to “LibreOffice” > “Java”, if you want to add further Java Runtime Environment versions.



4. Depending on the selected Java Runtime Environment version download the corresponding PostgreSQL JDBC driver.
5. Click “Class Path” to add the new driver.
6. Click “Add Archive” and select the driver.



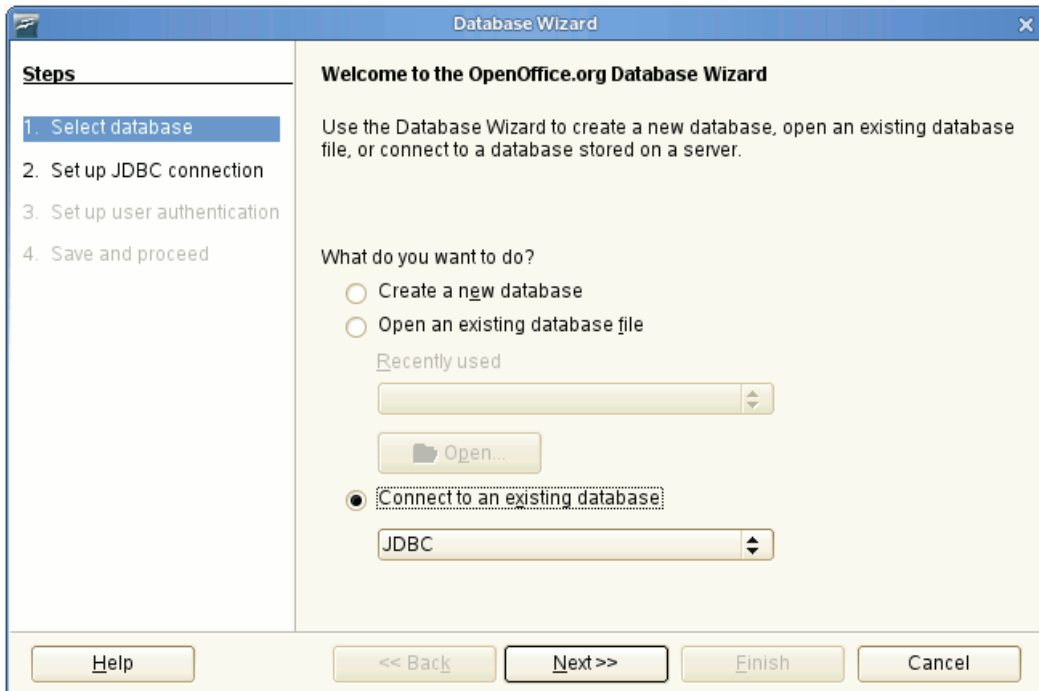
7. Confirm all dialog boxes and close LibreOffice Base. Verify that no LibreOffice process is running.

6.2.2 Connect from LibreOffice Base to the Database

To connect from LibreOffice to a PostgreSQL database, perform the following steps:

1. Start LibreOffice Base.

- In the LibreOffice Database Wizard select "Connect to an existing database" and "JDBC". Click "Next".



- Enter the data source URL and the JDBC driver class.

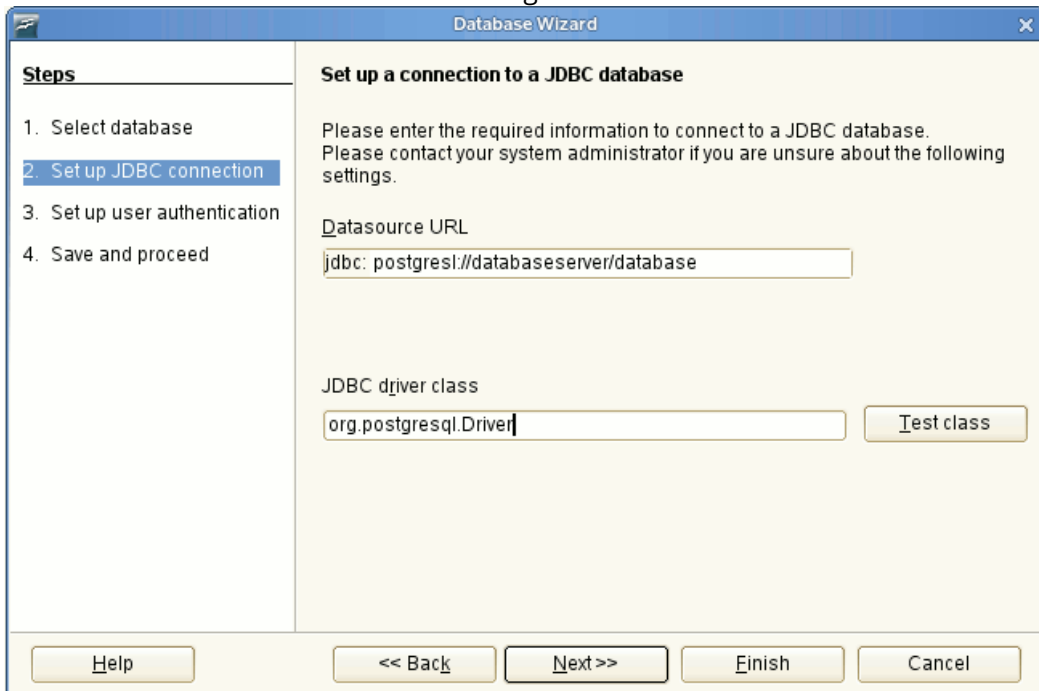
The data source URL has the following syntax:

`jdbc: postgresql://<fqdn of databaseserver>:<port>/<name of reporting database>`

Note: The port need not to be set if the default port (5432) is used.

The JDBC driver class is named `org.postgresql.Driver`.

Note: With the "Test class" button the settings can be tested.



- Click "Next".
- Enter user credentials and click "Finish".
- LibreOffice Base is started and queries and reports can be created.

