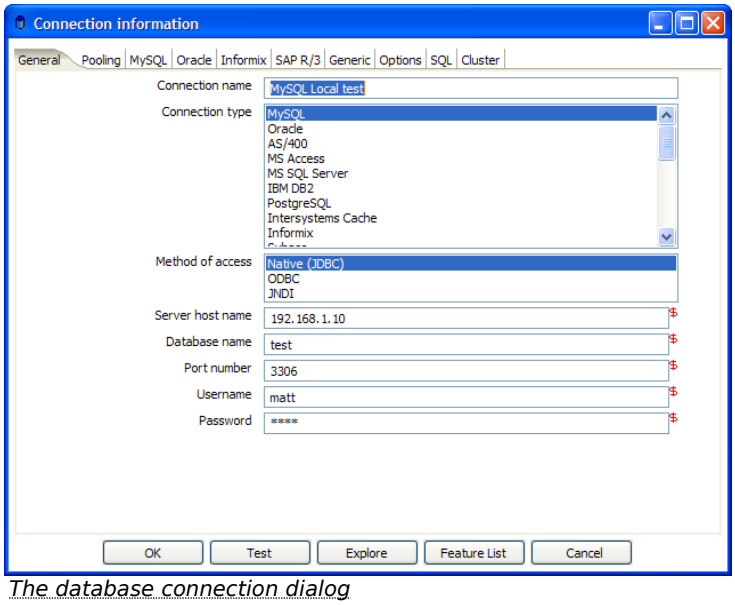
1 Database Connections

1.1. Description

A connection describes the method by which PME can connect to a database.

1.2. Screen shot



The database connection dialog

1.3. Options

Connection name: A connection name uniquely defines a connection.
Connection type: The type of database to which you are connecting.
Method of access: This can be Native (JDBC), ODBC, or OCI.
Server host name: Specify the host name of the server on which the database resides. You can also specify its IP address.
Database name: Identifies the database name to which you want to connect. In the case of ODBC, specify the DSN name here (see also 1.4. Database Usage Grid).
Port number: Sets the TCP/IP port number on which the database listens.
User name/password: Optionally specifies the user name and password to connect to the database.

EXTRA:

- For Oracle you can specify the default tablespace.
- For Informix, you need to specify the Informix Server name in the Informix tab in order for a connection to be usable.
- For SAP R/3 connections, extra parameters Language, System Number and SAP Client can be specified in the SAP R/3 tab.
- Feature list: Exposes the JDBC URL, class and various database settings for the connection such as the list of reserved words.
- Options: This new tab allows you to set database specific option on the database connections by adding parameters to the generated URL.

1.4. Database Usage Grid

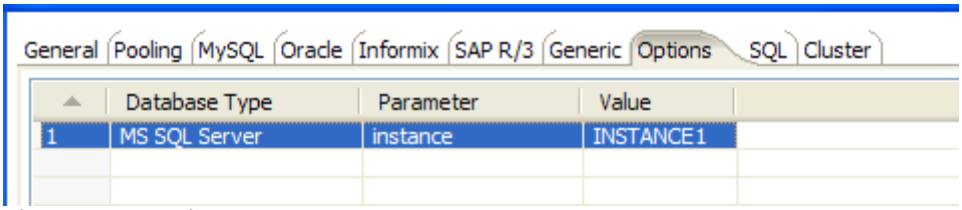
Database	Access Method	Server Name or IP Address	Database Name	Port # (default)	Username & Password
Oracle	Native	Required	Oracle database SID	Required (1521)	Required
	ODBC		ODBC DSN name		Required
	OCI		Database TNS name		Required
MySQL	Native	Required	MySQL database name	Optional (3306)	Optional
	ODBC		ODBC DSN name		Optional
AS/400	Native	Required	AS/400 Library name	Optional	Required
	ODBC		ODBC DSN name		Required
MS Access	ODBC		ODBC DSN name		Optional
MS SQL Server	Native	Required	Database name	Required (1433)	Required
	ODBC		ODBC DSN name		Required
IBM DB2	Native	Required	Database name	Required (50000)	Required
	ODBC		ODBC DSN name		Required
PostgreSQL	Native	Required	Database name	Required (5432)	Required
	ODBC		ODBC DSN name		Required
Intersystems Caché	Native	Required	Database name	Required (1972)	Required
	ODBC		ODBC DSN name		Required
Sybase	Native	Required	Database name	Required(5001)	Required
	ODBC		ODBC DSN name		Required
Gupta SQL Base	Native	Required	Database Name	Required (2155)	Required
	ODBC		ODBC DSN name		Required
Dbase III,IV or 5.0	ODBC		ODBC DSN name		Optional

Database	Access Method	Server Name or IP Address	Database Name	Port # (default)	Username & Password
Firebird SQL	Native	Required	Database name	Required (3050)	Required
	ODBC		ODBC DSN name		Required
Hypersonic	Native	Required	Database name	Required (9001)	Required
MaxDB (SAP DB)	Native	Required	Database name		Required
	ODBC		ODBC DSN name		Required
Ingres	Native	Required	Database name		Required
	ODBC		ODBC DSN name		Required
Borland Interbase	Native	Required	Database name	Required (3050)	Required
	ODBC		ODBC DSN name		Required
ExtenDB	Native	Required	Database name	Required (6453)	Required
	ODBC		ODBC DSN name		Required
Teradata	Native	Required	Database name		Required
	ODBC		ODBC DSN name		Required
Oracle RDB	Native	Required	Database name		Required
	ODBC		ODBC DSN name		Required
H2	Native	Required	Database name		Required
	ODBC		ODBC DSN name		Required
Netezza	Native	Required	Database name	Required (5480)	Required
	ODBC		ODBC DSN name		Required
IBM Universe	Native	Required	Database name		Required
	ODBC		ODBC DSN name		Required
SQLite	Native	Required	Database name		Required
	ODBC		ODBC DSN name		Required
Apache Derby	Native	optional	Database name	Optional (1527)	Optional
	ODBC		ODBC DSN name		Optional
Generic (*)	Native	Required	Database name	Required (Any)	Required
	ODBC		ODBC DSN name		Optional

(*) The generic database connection also needs to specify the URL and Driver class in the Generic tab!

1.5. MS SQL Server

To specify the SQL Server instance name, go to the Options tab and specify the parameter as shown:



The SQL Server "instance" property

To enable single sign-on login you can specify in a similar fashion the ***domain*** option:
From the jTDS FAQ on http://jtds.sourceforge.net/faq.html:

> *Specifies the Windows domain to authenticate in. If present and the user name and password are provided, jTDS uses Windows (NTLM) authentication instead of the usual SQL Server authentication (i.e. the user and password provided are the domain user and password). This allows non-Windows clients to log in to servers which are only configured to accept Windows authentication.*
>
> *If the `domain` parameter is present but no user name and password are provided, jTDS uses its native Single-Sign-On library and logs in with the logged Windows user's credentials (for this to work one would obviously need to be on Windows, logged into a domain, and also have the SSO library installed -- consult README.SSO in the distribution on how to do this).*

1.6. Oracle

If you do have issues with Oracle connectivity or other strange problems, you might want to consider replacing the Oracle JDBC driver to match your database server. Replace files “ojdbc14.jar” and “orai18n.jar” in the directory libext/JDBC of your distribution with the files found in the $ORACLE_HOME/jdbc directory on your server.

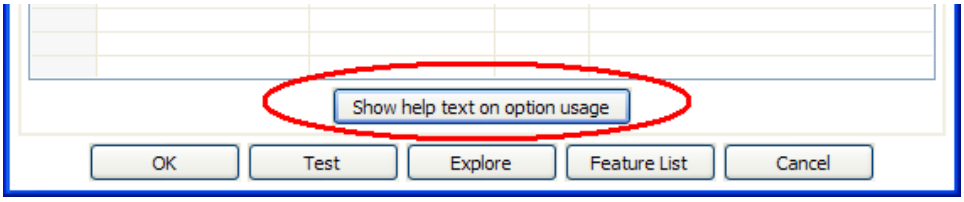
If you want to use OCI and an Oracle Net8 client, please read on. For OCI to work, the JDBC driver version used needs to match your Oracle client version. Oracle 2.5.0 shipped with version 10.1, 2.5.0 ships with version 10.2.
You can either install that version of the Oracle client or (probably easier) change the JDBC driver if versions don't match up. (see above)

1.7. MySQL

Because by default, MySQL gives back complete query results in one block to the client we had to enable “result streaming” by default. The big drawback of this is that it allows only 1 (one) single query to be opened at any given time. If you run into trouble because of that, you can disable this option in the MySQL tab of the database connection dialog.

Another issue you might come across is that the default timeout in the MySQL JDBC driver is set to 0. (no timeout) This leads to a problem in certain situations as it doesn't allow the detection of a server crash or sudden network failure if it happens in the middle of a query or open database connection. This in turn leads to infinite stalling. To solve this, set the “connectTimeout” and “socketTimeout” parameters for MySQL in the Options tab. The value to be specified is in milliseconds: for a 2 minute timeout you would specify value 120000 (2 x 60 x 1000).

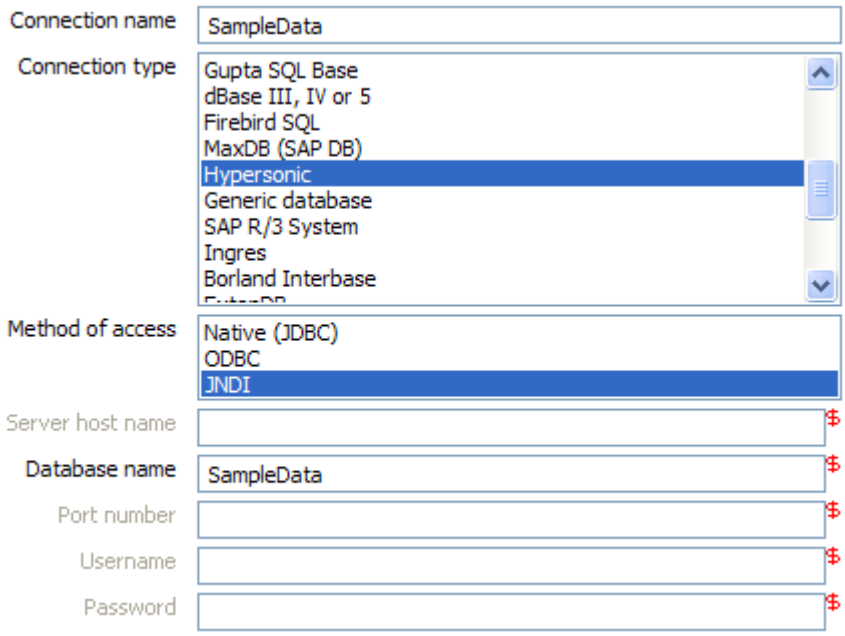
Also check out the other options on the linked MySQL help page. You can access this page using the supplied button.



1.8. JNDI

You can configure your database connections using JNDI. Because you don't want to have an application server running all the time during development or testing of the transformations, we have supplied a way of configuring a JNDI connection for “local” use. To configure, edit properties file called “simple-jndi/jdbc.properties.” For example, to connect to the databases used in Pentaho Demo platform download, use this information in the properties file:

```
SampleData/type=javax.sql.DataSource
SampleData/driver=org.hsqldb.jdbcDriver
SampleData/url=jdbc:hsqldb:hsql://localhost/sampledata
SampleData/user=pentaho_user
SampleData/password=password
Quartz/type=javax.sql.DataSource
Quartz/driver=org.hsqldb.jdbcDriver
Quartz/url=jdbc:hsqldb:hsql://localhost/quartz
Quartz/user=pentaho_user
Quartz/password=password
Hibernate/type=javax.sql.DataSource
Hibernate/driver=org.hsqldb.jdbcDriver
Hibernate/url=jdbc:hsqldb:hsql://localhost/hibernate
Hibernate/user=hibuser
Hibernate/password=password
Shark/type=javax.sql.DataSource
Shark/driver=org.hsqldb.jdbcDriver
Shark/url=jdbc:hsqldb:hsql://localhost/shark
Shark/user=sa
Shark/password=
```

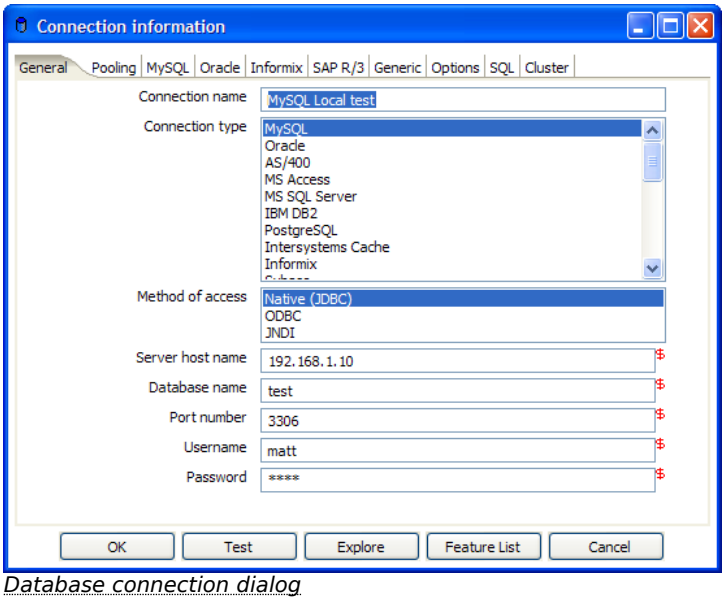


It is obviously important that the information stored in this file in the simple-jndi directory mirrors the content of your application server data sources.

1.9. Usage

1.9.1. Create a new connection

You can create a new connection by right clicking on the "Connections" tree entry and selecting "new".



Database connection dialog

1.9.2. General Tab

The general tab is where you setup the basic information about your connection like the connection name, type, access method, server name and login credentials.

1.9.2.1. Pooling Tab

The pooling tab allows you to configure your connection to use connection pooling and define options related to connection pooling like the initial pool size, maximum pool size and connection pool parameters.

1.9.2.2. Options Tab

This tab displays a variety of configurable, data source specific options (i.e. defaultFetchSize, useCursorFetch, etc.). Clicking on the 'Show help text on option usage' button will launch a browser window.

1.9.2.3. SQL Tab

This tab allows you to enter a number of SQL commands immediately after connecting to the database. This is sometimes needed for various reasons like licensing, configuration, logging, tracing, etc.

1.9.2.4. Cluster Tab

This tab allows you to enable clustering for the database connection and create connections to the data partitions. To create a new data partition, enter a partition ID and the hostname, port, database, username and password for connecting to the partition.

1.9.3. Edit a connection

Double click on the connection name in the left tree. Or right click on the name and select "Edit connection".

1.9.4. Duplicate a connection

Right click on the connection name and select "Duplicate".

1.9.5. Copy to clipboard

Copies the XML describing the connection to the clipboard.

1.9.6. Delete a connection

Right click on the connection name and select "Delete".

1.9.7. Test a connection

In the edit window (see above), select the "Test" button. If connection succeeds, an OK message is displayed after a short delay.

1.9.8. Execute SQL commands on a connection

Right click on the connection name and select "SQL Editor".

1.9.9. Clear DB Cache option

To speed up connections, a database cache is used. Use this option when the information in the cache no longer represents the layout of the database. This is the case when databases tables have been changed, created or deleted.

1.9.10. Explore

This option will start the database explorer for the selected database connection.

1.10. Unsupported databases

If you want to access a database type that is not yet supported, let us know and we will try to find a solution. A few database types are not supported in this release because of the lack of sample database and/or software.

Please note that it is usually still possible to read from these databases by using the Generic database driver through an ODBC or JDBC connection.