

SQLdb Package 编辑

当前位置： [文江博客](#) 知识库 [Free Pascal SQLdb](#)

The SQLdb package contains FPC units to access a number of SQL databases. It is "packaged" as sqldb.laz in Lazarus, and the components can be found on the [SQLdb tab](#).



Documentation

See [SQLDB Documentation](#)

Components

The SQLdb package provides the following components:

TXXXConnection

Documentation: see [TSQLConnection documentation](#)

There are various connection components called TXXXConnection, where XXX is the flavour of the database you are connecting to. Each one of these components takes the requests of the SQLQuery and SQLTransaction components and translates them into requests specifically tailored for the database you are using. These connection objects descend from the generic TSQLConnection; you can use TSQLConnection to create database programs that can connect to multiple databases (see [SQLdb Tutorial3](#)).

The actual components are:

- TIBConnection ([Borland Interbase / Firebird](#))
- TMSSQLConnection ([Microsoft SQL Server](#), available since FPC version 2.6.1/Lazarus 1.0.8)
- TMySQL40Connection ([MySQL](#), requires MySQL 4.0 client library)
- TMySQL41Connection ([MySQL](#), requires MySQL 4.1 client library)
- TMySQL50Connection ([MySQL](#), requires MySQL 5.0 client library)
- TMySQL51Connection ([MySQL](#), requires MySQL 5.1 client library, available since FPC version 2.5.1)
- TMySQL55Connection ([MySQL](#), requires MySQL 5.5 client library, available since FPC version 2.6.1/Lazarus 1.0.8)
- TODBCConnection (An ODBC connection to a database that the PC has the driver for; see [ODBCConn](#))

- TOracleConnection ([Oracle](#))
- TPQConnection ([PostgreSQL](#))
- TSybaseConnection ([Sybase ASE](#), available since FPC version 2.6.1/Lazarus 1.0.8)
- TSQLite3Connection ([SQLite](#), available since FPC version 2.2.2)
- TMSSQLConnection ([MSSQL](#), requires FreeTDS library)

TSQLTransaction

Documentation: see [TSQLTransaction documentation](#)

This encapsulates the transaction on the database server. A TXXXConnection object always needs at least one TSQLTransaction associated with it, so that the transaction of its queries is managed.

Queries/actions on the database need to be encapsulated in

- a .StartTransaction/.Commit (or .Rollback) block
- or a single .StartTransaction at the beginning, followed by .CommitRetaining or .RollbackRetaining. CommitRetaining and RollbackRetaining keep the transaction open, so performing another StartTransaction is not needed.

Setting .Active to true if it was false is equivalent to .StartTransaction.

Setting .Active to false if it was true is equivalent to .Rollback

TSQLQuery

Documentation: see [TSQLQuery](#)

This is a descendant of TDataset, and provides the data as a table from the SQL query that you submit. However, it can also be used to execute SQL queries (e.g. stored procedures, INSERT INTO..) that don't return any data. See [Working With TSQLQuery](#) for more details.

Using the SQLdb Package

To use the SQLdb package, you need a TSQLQuery component, a TSQLTransaction component, and one of the connection components.

Here is a quick tutorial:

- Make sure you have the relevant database client/driver installed.
- Go to the SQLdb tab in Component Palette.
- Add a TXXXConnection component (suitable for the type of database you are using) to a form, and fill in the appropriate properties to connect to your database. These will vary depending on the actual database. Make sure you can set the "connected" property to true.
- Add a TSQLTransaction component. Return to the Connection component, and set its Transaction Property to the new transaction. This should also set the transaction's database property. You should be able to set active to true.

- Add an SQLQuery component and set its Database property to the connecton component you just added.
- Set the SQLQuery "SQL" property. For a first test, just do a "select * from <tablename>".
- Set active to true. If this works, you have opened your query, and the data is available.

From here on, adding controls is the same as for any database, but just to recap:

- Add a TDataSource component (from the Data Access tab) and set its dataset property to the SQLQuery component
- Add some controls from the Data Controls tab. For a quick check, add a TDBGrid component, and set its datasource property to the Datasource component you just added. You should see the data from the query you entered.

Of course, this is just the beginning, but getting some data in a grid is usually the first step to ensure the database connection is working. More detailed explanations will be found at [Working With TSQLQuery](#) and [SQLdb Tutorial1](#).

See also

收藏 0 已收藏 0



分享到微信



[分享到QQ](#)

[分享到微博](#)