Lazarus Database Overview 编辑

Databases portal

References:

- General info
- Libraries
- Field types
- Controls
- FAQ
- SQL how-to
- Working With TSQLQuery
- In-memory database applications

Tutorials/practical articles:

- Overview
- 0 - Database set-up
- 1 - Getting started
- 2 - Editing
- 3 - Queries
- 4 - Data modules
- SQLdb Programming Reference

Databases

Advantage - MySQL - MSSQL - Postgres - Interbase - Firebird - Oracle - ODBC - Paradox - SQLite - dBASE - MS Access - Zeos

Overview

This article is an overview of which databases can work with Lazarus.

Lazarus supports several databases out of the box (using e.g. the SQLDB framework), however the developer must install the required packages (client libraries) for each one.

You can access the database through code or by dropping components on a form. The data-aware components represent fields and are connected by setting the DataSource property to point to a TDataSource. The Datasource represents a table and is connected to the database components (examples: *TPSQLDatabase*, *TSQLiteDataSet*) by setting the DataSet property. The data-aware components are located on the Data Controls tab. The Datasource and the database controls are located on the "Data Access" tab.

See the tutorials for Lazarus/FPC built in database access, suitable for Firebird, MySQL, SQLite, PostgreSQL etc:

Lazarus and Interbase / Firebird

- Firebird is very well supported out of the box by FPC/Lazarus (using SQLDB); please see Firebird for details.
- Other Firebird libraries has a list of alternative access libraries (e.g. PDO, Zeos, FBlib)

Lazarus and MySQL

- Please see mysql for details on various access methods, which include:

1. Built-in SQLdb support
2. PDO
3. Zeos
4. MySQL data access Lazarus components

Lazarus and MSSQL/Sybase

You can connect to Microsoft SQL Server databases using

1. SQL Server data access Lazarus components.They are working on Windows and macOS. Free to download.
2. The built-in **SQLdb** connectors **TMSSQLConnection** and **TSybaseConnection** (since Lazarus 1.0.8/FPC 2.6.2): see mssqlconn.
3. **Zeos** component **TZConnection** (latest CVS, see links to Zeos elsewhere on this page)
    1. On Windows you can choose between native library **ntwdblib.dll** (protocol **mssql**) or FreeTDS libraries (protocol **FreeTDS_MsSQL-nnnn**) where nnnn is one of four variants depending on the server version. For Delphi (not Lazarus) there is also another Zeos protocol **ado** for MSSQL 2005 or later. Using protocols mssql or ado generates code not platform independient.
    2. On Linux the only way is with FreeTDS protocols and libraries (you should use **libsybdb.so**).
4. **ODBC** (MSSQL and Sybase ASE) with SQLdb **TODBCConnection** (consider using **TMSSQLConnection** and **TSybaseConnection** instead)
    1. See also [1]
    2. On Windows it uses native ODBC Microsoft libraries (like sqlsrv32.dll for MSSQL 2000)
    3. On Linux it uses unixODBC + FreeTDS (packages unixodbc or iodbc, and tdsodbc). Since 2012 there is also a Microsoft SQL Server ODBC Driver 1.0 for Linux which is a binary product (no open source) and provides native connectivity, but was released only for 64 bits and only for RedHat.

Lazarus and ODBC

ODBC is a general database connection standard which is available on Linux, Windows and macOS. You will need an ODBC driver from your database vendor and set up an ODBC "data source" (also known as

DSN). You can use the SQLDB components (TODBCConnection) to connect to an ODBC data soruce. See ODBCConn for more details and examples.

Microsoft Access

You can use the ODBC driver on Windows as well as Linux to access Access databases; see MS Access

Lazarus and Oracle

- See Oracle. Access methods include:

1. Built-in SQLDB support
2. Zeos
3. Oracle data access Lazarus component

Lazarus and PostgreSQL

- PostgreSQL is very well supported out of the box by FPC/Lazarus
- Please see postgres for details on various access methods, which include:

1. Built-in SQLdb support. Use component **TPQConnection** from the SQLdb tab of the Component Palette
2. Zeos. Use component **TZConnection** with protocol 'postgresql' from palette **Zeos Access**
3. PostgreSQL data access Lazarus component

Lazarus and SQLite

SQLite is an embedded database; the database code can be distributed as a library (.dll/.so/.dylib) with your application to make it self-contained (comparable to Firebird embedded). SQLite is quite popular due to its relative simplicity, speed, small size and cross-platform support.

Please see the SQLite page for details on various access methods, which include:

1. Built-in SQLDb support. Use component **TSQLite3Connection** from palette **SQLdb**
2. Zeos
3. SQLitePass
4. TSQLite3Dataset
5. SQLite data access Lazarus components

Lazarus and Firebird/Interbase

InterBase (and FireBird) Data Access Components (IBDAC) is a library of components that provides native connectivity to InterBase, Firebird and Yaffil from Lazarus (and Free Pascal) on Windows, macOS, iOS, Android, Linux, and FreeBSD for both 32-bit and 64-bit platforms. IBDAC-based applications connect to the server directly using the InterBase client. IBDAC is designed to help programmers develop faster and cleaner InterBase database applications.

IBDAC is a complete replacement for standard InterBase connectivity solutions. It presents an efficient alternative to InterBase Express Components, the Borland Database Engine (BDE), and the standard dbExpress driver for access to InterBase.

Firebird data access components for Lazarus are free to download.

Lazarus and dBase

FPC includes a simple database component that is derived from the Delphi TTable component called "TDbf" TDbf Website). It supports various DBase and Foxpro formats.

TDbf does not accept SQL commands but you can use the dataset methods etc and you can also use regular databound controls such as the DBGrid.

It doesn't require any sort of runtime database engine. However it's not the best option for large database applications.

See the TDbf Tutorial page for the tutorial as well as documentation.

You can use e.g. OpenOffice/LibreOffice Base to visually create/edit dbf files, or create DBFs in code using TDbf.

Lazarus and Paradox

Paradox was the default format for database files in old versions of Delphi. The concept is similar to DBase files/DBFs, where the "database" is a folder, and each table is a file inside that folder. Also, each index is a file too. To access this files from Lazarus we have these options:

- TParadox: Install package "lazparadox 0.0" included in the standard distribution. When you install this package, you will see a new component labeled "PDX" in the "Data Access" palette. This component is not standalone, it uses a "native" library, namely the pdxlib library which is available for Linux and Windows. For example, to install in Debian, you could get **pxlib1** from package manager. In Windows you need the pxlib.dll file.
- TPdx: Paradox DataSet for Lazarus and Delphi from this site. This component is standalone (pure object pascal), not requiring any external library, but it can only read (not write) Paradox files. The package to install is "paradoxlaz.lpk" and the component should appear in the "Data Access" palette with PDX label (but orange colour).
- TParadoxDataSet: is a TDataSet that can only read Paradox Files up to Version 7. The approach is similar to the TPdx component, the package to install is "lazparadox.lpk" and the component should also appear in the "Data Access" palette.

TSdfDataset and TFixedDataset

TSdfDataSet and TFixedFormatDataSet are two simple TDataSet descandants which offer a very simple textual storage format. These datasets are very convenient for small databases, because they are fully

implemented as an Object Pascal unit, and thus require no external libraries. Also, their textual format allows them to be easily viewed/edited with a text editor.

See CSV for example code.

Lazarus and Advantage Database Server

- Please see Advantage Database Server for details on using Advantage Database Server

See also

(Sorted alphabetically)

External links

- Pascal Data Objects - a database API that worked for both FPC and Delphi and utilises native MySQL libraries for version 4.1 and 5.0 and Firebird SQL 1.5, and 2.0. It's inspired by PHP's PDO class.
- Zeos+SQLite Tutorial - Good tutorial using screenshots and screencasts it explain how to use SQLite and Zeos, spanish (google translate does a good work in translating it to english)