



JDBC Extension Library

TECHNICAL REFERENCE

Version 0.1

Table of Contents

Introduction	3
JDBC Classes	3
Accessing Databases	3
Parameters	4
Transactions	4
Sample Code.....	4
Dependencies.....	5
License.....	5
Obtaining	5

Introduction

The **JDBC Extension Library for .NET** implements an ADO.NET-JDBC bridge that allows .NET applications to access databases over JDBC through a Java host process.

You will find the code documentation at

<https://robbiblubber.lima-city.de/Robbiblubber.Extend.SQL.JDBC.NET/>.

JDBC Interop documentation is available at

<https://bitbucket.org/robbiblubber/robbiblubber.extend.sql.jdbc.interop.java/downloads/Robbiblubber.Extend.SQL.JDBC.Interop.pdf>.

JDBC Classes

The JDBC Extension Library provides implementations of the basic ADO.NET interfaces. These implementations do not support more than one transaction at a time.

The core implementation contains the following classes:

```
JdbcCommand  
JdbcConnection  
JdbcDataAdapter  
JdbcParameter  
JdbcParameterCollection  
JdbcReader  
JdbcTransaction
```

The `JdbcProvider` class allows integration with the robbiblubber.org SQL Framework.

Accessing Databases

To create a JDBC connection you will want to create a `JdbcConnection` object. The [constructor](#) allows you to specify the **connection string** (URL in JDBC), user name and password, and a driver class. A **database driver class** is required for all JDBC drivers because no drivers are bound to the host at compile time.

All Java resources necessary for the host process to load a requested driver must be present in the application directory, most likely in the `lib/` directory. The class loader will automatically detect and load the specified driver.

Parameters

The JDBC Extension library supports **named parameters** typically used in ADO.NET. It does not support output parameters.

Transactions

The JDBC Extension library supports a **single transaction** at a time. This transaction is started with the `JdbcConnection.StartTransaction()` method.

Sample Code

The following code example shows how to access a SQLite database via JDBC Extension Library. The SQLite JDBC driver must be present in the application directory.

```
JdbcConnection cn = new JdbcConnection("jdbc:sqlite:C:/db.sqlite",
                                       "org.sqlite.JDBC");

JdbcCommand cmd = cn.CreateCommand("INSERT INTO TAB2 VALUES 'foo'");
cmd.ExecuteNonQuery();

cmd = cn.CreateCommand("SELECT * FROM TAB2 WHERE CV = :bar");
cmd.Parameters.Add(new JdbcParameter(":bar", "BAR"));

JdbcReader re = cmd.ExecuteReader();

while(re.Read())
{
    Console.WriteLine(re.GetString(0));
}
re.Close();

cn.Close();
```

SQLite database access sample.

Dependencies

The JDBC Interop API uses the robbiblubber.org Utility Suite for Java, the robbiblubber.org SQL Framework, and the robbiblubber.org Named Pipes Extension Library..

License

The JDBC Interop API and the JDBC Interop server are under MIT License. You can get a copy at <https://bitbucket.org/robbiblubber/robbiblubber.extend.sql.jdbc.net/downloads/License.txt>.

Obtaining

The complete sources are available at <https://bitbucket.org/robbiblubber/robbiblubber.extend.sql.jdbc.interop.net/>.