

Configuring DB2 datasource

{scrollbar}

top

This article shows you how to configure a DB2 datasource in Apache Geronimo v1.2. Since the release of Geronimo v1.1 you can now select multiple drivers from the creation pool wizard on the Geronimo Administration Console.

In this scenario we are creating a connection pool for a DB2 database, in order to connect to that database we will need to define at least two driver jar files. That is the JDBC driver itself and the respective license files. Depending on your implementation you will need different license files.

In this article we provide two alternatives to deploy the connection pool, the first approach we tackle is using the Geronimo Administration Console, then we explain the process for using the command line option instead.

Add DB2 drivers to the repository

To add the DB2 drivers and license to the Geronimo repository you will need to place those files in a particular directory structure. You will normally have two or three files to add to the repository, these files are:

- **db2jcc.jar** - This is the actual DB2 Universal JDBC Driver jar file.
- **db2jcc_license_cu.jar** - This is the standard DB2 Universal JDBC driver license file that allows access to the DB2 Universal database for Linux, UNIX and Windows servers.
- **db2jcc_license_cisuz.jar** - This is the DB2 (DB2 ESE and DB2 Connect) JDBC Driver license for z/OS and iSeries that should be used in addition to the standard license for these servers.

These files are available in the <sqlib_home>\java directory. For additional information on the DB2 JDBC drivers and licenses visit the DB2 Information Center available at the following URL:

<http://publib.boulder.ibm.com/infocenter/db2luw/v8/topic/com.ibm.db2.udb.doc/ad/t0010264.htm>

In order to use these files in Geronimo you will need to rename (copy and rename) as described in the following table.

Original name	Renamed
db2jcc.jar	db2jcc-8.1.8.jar
db2jcc_license_cu.jar	db2jcc_license_cu-8.1.8.jar
db2jcc_license_cisuz.jar	db2jcc_license_cisuz-8.1.8.jar

This means that for this particular case DB2 v8.1.8.806 was used.

At this point you are ready to add those files to the Geronimo repository, to do that you have two alternatives. You can either use any graphical or command line tool to copy the files and create the necessary directories or you use the Geronimo Administration Console and add the driver and licenses to the common libraries. Next we will cover both alternatives.

Using command line

You will need to create the following directory structures under the <geronimo_home>\repository directory and copy the appropriate files to the respective directories.

- **com/ibm/db2/db2jcc/8.1.8**
and copy the **db2jcc-8.1.8.jar** into that directory.
- **com/ibm/db2/db2jcc_license_cisuz/8.1.8**
and copy the **db2jcc_license_cisuz-8.1.8.jar** into that directory.
- **com/ibm/db2/db2jcc_license_cu/8.1.8**
and copy the **db2jcc_license_cisuz-8.1.8.jar** into that directory.

Using the Geronimo Administration Console

In order to use the console Apache Geronimo must be running. Access the Geronimo Administration Console by pointing your browser to the following URL:

<http://localhost:8080/console>

- Enter the **system** as the user and **manager** as the password and click **Login**.
- Click on **Common Libs** to access the **Repository viewer** portlet.
- Click on **Browse** and select the first file to install. In this case we will first install **db2jcc-8.1.8.jar**.
- A set of values will be proposed by default, set the **Group** to **com.ibm.db2**, leave the rest by default and click **Install**.
- Repeat the previous two steps for **db2jcc_license_cisuz-8.1.8.jar** and **db2jcc_license_cisuz-8.1.8.jar**

With the drivers and license files installed you can now create a new database connection pool.

Create a database connection pool using the wizard from the Geronimo Administration Console

From the Geronimo Administration Console select **Database Pools** and create a new pool by clicking on **Using the Geronimo database pool wizard**.

On Step 1 of the create database pool wizard enter the pool name and database type as illustrated in the following figure and then click **Next**.

The screenshot shows a web-based wizard titled "Database Pools" with a "[view]" link in the top right corner. The main heading is "Create Database Pool -- Step 1: Select Name and Database". There are two input fields: "Name of Database Pool:" with the text "DB2_ds" entered, and "Database Type:" with "DB2" selected from a dropdown menu. Below the "Name of Database Pool:" field is a note: "A name that is different than the name for any other database pools in the server (no spaces in the name please).". Below the "Database Type:" field is a note: "The type of database the pool will connect to.". At the bottom left is a "Cancel" link, and at the bottom center is a "Next" button.

On Step 2 leave the **JDBC Driver Class:** field by default (com.ibm.db2.jcc.DB2Driver). If you click on the **Driver JAR:** pull-down menu you should now see the three com.ibm/db2... jars you copied in the repository. Since Geronimo v1.1 only allow you to select ONE jar file from this menu select the first jar on the list, **com.ibm/db2jcc/8.1.8/jar** which is the driver itself. You will add manually the license jars in the next steps.

Enter the remaining connection information as shown in the following figure. For this example the default **db2admin** user and password was used and a **SAMPLE** database was created via the DB2 Control Center.

Create Database Pool -- Step 2: Select Driver, JAR, Parameters

JDBC Driver Class:

See the documentation for your JDBC driver.

Driver JAR:

axis/axis/1.4/jar
backport-util-concurrent/backport-util-concurrent/2.2/jar
castor/castor/0.9.5.3/jar
com.ibm.db2/db2jcc/8.1.8/jar
com.ibm.db2/db2jcc_license_cisuz/8.1.8/jar
com.ibm.db2/db2jcc_license_cu/8.1.8/jar
concurrent/concurrent/1.3.4/jar
console.dbpool/DefaultDS/1.0/rar
console.dbpool/examples-dataSource-demoPool/1.0/rar
dwr/dwr/1.1.1/jar

The JAR(s) required to make a connection to the database. Use CTRL-click or SHIFT-click to select multiple jars.

The JAR(s) should already be installed under Geronimo/repository/ (or [Download a Driver](#))**DB User Name:**

The username used to connect to the database

DB Password: **Confirm Password:**

The password used to connect to the database

Driver Connection Properties**Typical JDBC URL:** **Port:**

A property used to connect to DB2. May be optional (see JDBC driver documentation).

Host:

A property used to connect to DB2. May be optional (see JDBC driver documentation).

Database:

A property used to connect to DB2. May be optional (see JDBC driver documentation).

[Cancel](#)Click **Next**.Leave the options on Step 3 by default, you should see the message **Driver Status: Loaded Successfully**.

Database Pools
[view]

Create Database Pool -- Step 3: Final Pool Configuration

JDBC Connect URL: jdbc:db2://localhost:50000/SAMPLE

Make sure the generated URL fits the syntax for your JDBC driver.

Driver Status: Loaded Successfully

Connection Pool Parameters

Pool Min Size:

The minimum number of connections in the pool. Leave blank for default.

Pool Max Size:

The maximum number of connections in the pool. Leave blank for default.

Blocking Timeout: (in milliseconds)

The length of time a caller will wait for a connection. Leave blank for default.

Idle Timeout: (in minutes)

How long a connection can be idle before being closed. Leave blank for default.

Test Connection Skip Test and Deploy Skip Test and Show Plan

Cancel

Click on **Test Connection**, you should see a confirmation message that you are connected to DB2.

Database Pools
[view]

Create Database Pool -- Step 4: Test Connection

Test Result: Connected to DB2/NT SQL08021

Deploy Show Plan

Cancel

Click on **Deploy**. You should now have **DB2_ds** listed in the **Database Pools** portlet.

Database Pools
[view]

This page lists all the available database pools.

For each pool listed, you can click the **usage** link to see examples of how to use the pool from your application.

Name	Deployed As	State	Actions
DB2_ds	Server-wide	running	edit usage
DefaultDS	Server-wide	running	edit usage
NoTxDataSource	Server-wide	running	edit usage
SystemDataSource	Server-wide	running	edit usage
examples-datasource-demoPool	Server-wide	running	edit usage

Create a new database pool:

- [Using the Geronimo database pool wizard](#)
- [Import from JBoss 4](#)
- [Import from WebLogic 8.1](#)

Deploy a database conneciton pool using the command line

As an alternatively to the wizard you could create a deployment plan manually and deploy it using the command line based deployer tool. To use this option create a **db2-plan.xml** file and copy the content of the following example.

```

xmldbb2-plan.xml deployment plan <?xml version="1.0" encoding="UTF-8"?> <connector xmlns="http://geronimo.apache.org/xml/ns/j2ee/connector-1.2"> <dep:environment xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.2"> <dep:moduleId> <dep:groupId>console.dbpool</dep:groupId> <dep:artifactId>DB2_ds</dep:artifactId> <dep:version>1.0</dep:version> <dep:type>rar</dep:type> </dep:moduleId> <dep:dependencies> <dep:dependency> <dep:groupId>com.ibm.db2</dep:groupId> <dep:artifactId>db2jcc</dep:artifactId> <dep:version>8.1.8</dep:version> <dep:type>jar</dep:type> </dep:dependency> <dep:dependency> <dep:groupId>com.ibm.db2</dep:groupId> <dep:artifactId>db2jcc_license_cisuz</dep:artifactId> <dep:

```

```

version>8.1.8</dep:version> <dep:type>jar</dep:type> </dep:dependency> <dep:dependency> <dep:groupId>com.ibm.db2</dep:groupId> <dep:
artifactId>db2jcc_license_cu</dep:artifactId> <dep:version>8.1.8</dep:version> <dep:type>jar</dep:type> </dep:dependency> </dep:dependencies> <
/dep:environment> <resourceadapter> <outbound-resourceadapter> <connection-definition> <connectionfactory-interface>javax.sql.DataSource<
/connectionfactory-interface> <connectiondefinition-instance> <name>DB2_ds</name> <config-property-setting name="Password">db2admin</config-
property-setting> <config-property-setting name="Driver">com.ibm.db2.jcc.DB2Driver</config-property-setting> <config-property-setting name="
UserName">db2admin</config-property-setting> <config-property-setting name="ConnectionURL">jdbc:db2://localhost:50000/SAMPLE</config-property-
setting> <connectionmanager> <local-transaction/> <single-pool> <max-size>10</max-size> <min-size>0</min-size> <match-one/> </single-pool> <
/connectionmanager> </connectiondefinition-instance> </connection-definition> </outbound-resourceadapter> </resourceadapter> </connector>

```

Let's analyze this plan now. Take a look at the **<dep:environment>** section, there you can find the `moduleId` which identify the resource or component being deployed; in the Administration Console the `moduleId` is displayed in the **Component Name** column on the **Database Pools** portlet.

Right after the `moduleId` comes the definition of the dependencies. In this particular case you can find three **<dep:dependency>** blocks pertaining to the DB2 JDBC driver and the two license jars. The last "big" block in this plan is the **<resourceadapter>** where the connection parameters such as driver, user and password, connection URL, etc. are defined.

Deploy the datasource

To the deploy the DB2 datasource you just created run the following command from the `<geronimo_home>\bin` directory.

```

deploy --user system --password manager deploy <dep_plan_home>\db2-plan.xml ..\repository\org\tranql\tranql-
connector-ra\1.3\tranql-connector-ra-1.3.rar

```

You should receive the following message:

[Back to Top](#)