

DB2V7

DB2 Datasources

To use DB2 databases in Version 7 (or above?) from Cocoon you have to complete the following steps:

- have DB2 support available to Cocoon
 - make the JDBC driver available to Cocoon
 - configure a DB2 datasource
 - use the datasource from esql logicsheet or sql-transformer
-

DB2 support for Cocoon

Here you have to install DB2, to catalog your databases in the database directory, to set the environment (usually using the command `db2profile`), to switch to JAVA2 support (usually using the command `.../sqlib/java12/usejdbc2`).

Now you should be able to connect to a DB2 database from the command line with a command like "db2 connect to mydatabase user myuserid".

Make the JDBC driver available to Cocoon

If DB2 is installed correctly, then the CLASSPATH is already extended to the DB2 directories.

Now you have to assure that your servlet server uses these classpath settings. For Tomcat you have to modify the command `.../bin/setclasspath.sh` (or `.bat`). The default of Tomcat is not to use the classpath settings at all!

```
# Set standard CLASSPATH
CLASSPATH=$CLASSPATH:"$JAVA_HOME"/lib/tools.jar
```

Here the original classpath has been added before `tools.jar`

Now add the DB2 JDBC driver to the list of preloaded classes in `web.xml`:

```
<init-param>
  <param-name>load-class</param-name>
  <param-value>
    <!-- For IBM WebSphere:
    com.ibm.servlet.classloader.Handler -->

    <!-- For Database Driver: -->
    org.hsqldb.jdbcDriver

    oracle.jdbc.driver.OracleDriver
    COM.ibm.db2.jdbc.app.DB2Driver

    <!-- For parent ComponentManager sample:
    org.apache.cocoon.samples.parentcm.Configurator
    -->
  </param-value>
</init-param>
```

In this example the Oracle driver and the DB2 driver is added.

Configure a DB2 datasource

To configure a DB2 datasource you have to add a `jdbc` element to `cocoon.xconf`:

```

<jdbc name="info">
  <pool-controller min="1" max="30" auto-commit="false">
    <keep-alive disable="true"/>
  </pool-controller>
  <dburl>jdbc:db2:INFO</dburl>
  <user>uid</user>
  <password>pwd</password>
</jdbc>

```

This example defines the datasource named "info".

You should always set keep-alive/disable to true, because the default sends an invalid SQL statement to the database with degrades performance! (You can find many errors "SELECT 1" in the db2diag.log on your database server)

If you want to have "real" transactions don't forget to set auto-commit to false!

Use the datasource from esql logicsheet

To use the datasource from a xsp serverpage with esql you use something like this:

```

<esql:connection>
  <esql:pool>Info</esql:pool>
  <esql:autocommit>false</esql:autocommit>

  <!-- Select something ... -->
  <esql:execute-query>
    <esql:query>
      select x, y,z
      from some.table t
      where x < 7
    </esql:query>
    <esql:results>
      <esql:row-results>
        <Something><esql:get-columns/></Something>
      </esql:row-results>
    </esql:results>
  </esql:execute-query>

  <!-- Delete something ... -->
  <esql:execute-query>
    <esql:query>
      delete from some.table t
      where x = 4
    </esql:query>
  </esql:execute-query>

  <!-- Commit it ... -->
  <esql:execute-query>
    <esql:query>
      commit
    </esql:query>
  </esql:execute-query>

</esql:connection>

```

If you don't want to have committed every single statement you have to state autocommit=false on every esql:connection!

Use the datasource from sql-transformer

((to be continued))

and thats all!

[ChristophOberle](#)