Apache Roller on Geronimo (1.2 Ok)

This article focuses on configuring Apache Roller v3.1 on the Apache Geronimo v1.2 server. Roller is a powerful and popular open source blog server and this document will walk you throw the configuration steps needed to get Roller up and running with the Derby or MySql database back-end on Geronimo. For detailed information on the usage of JDBC in Geronimo, refer the Simple database access sample application (1.2 Ok) article.

After reading this article you should be able to configure the Geronimo application server for the Apache Roller application with any of the Roller supported databases back-ends.

Apache Roller resources

If you do not already have access to the roller v3.1 application here is:

- The bundle: apache-roller-3.1
- Required jars: required-jars-roller-3.1

Included in the apache roller bundle above is database scripts for db2, derby, hsqldb, mysql, mssql and oracle. You will find the scripts in apache-roller /webapp/roller/WEB-INF/dbscripts/

Configuring, Building and Deploying the Apache Roller application

Make sure you read the roller-install-guide for detailed instructions on setting up Apache Roller. The following is mostly Geronimo specific configuration instructions.

Preparing the resources

If you are planing on experimenting with or later upgrade this Roller Geronimo setup you should probably let ant or Maven handle most of the setup but for simplicity this document will use a manual approach.

- Unpack the **apache-roller-3.1** and **required-jars-roller-3.1** into a common working directory (make sure the required jars end up in the roller */WEB-INF/lib* directory)
- Place the **geronimo-web.xml** (see below) alongside with the web.xml file in the *roller/WEB-INF/* directory
- Place the **roller-custom.properties** (see below) file alongside with the roller.properties file in the roller */WEB-INF/classes* directory.

You will later re-pack the *roller/...* directory into a roller.war file but first we need to set up the database.

geronimo-web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://geronimo.apache.org/xml/ns/j2ee/web-1.2"
xmlns:nam="http://geronimo.apache.org/xml/ns/naming-1.1"
xmlns:sec="http://geronimo.apache.org/xml/ns/security-1.1"
xmlns:sys="http://geronimo.apache.org/xml/ns/deployment-1.1">
  <sys:environment>
    <sys:moduleId>
      <sys:groupId>roller</sys:groupId>
      <sys:artifactId>roller</sys:artifactId>
      <sys:version>3.1</sys:version>
      <sys:type>war</sys:type>
    </sys:moduleId>
    <sys:dependencies>
      <sys:dependency>
         <sys:groupId>console.dbpool</sys:groupId>
         <sys:artifactId>rollerdb</sys:artifactId>
      </sys:dependency>
    </sys:dependencies>
    <hidden-classes>
        <filter>antlr</filter>
        <filter>org.apache.commons.digester</filter>
    </hidden-classes>
  </sys:environment>

  <context-root>/roller</context-root>

  <sys:resource-ref>
     <sys:ref-name>jdbc/rollerdb</sys:ref-name>
     <sys:resource-link>rollerdb</sys:resource-link>
  </sys:resource-ref>

    <gbean name="rollerdb_sr" class="org.apache.geronimo.security.realm.GenericSecurityRealm" xsi:type="dep:
gbeanType"
           xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.2"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <attribute name="realmName">rollerdb_sr</attribute>
        <reference name="ServerInfo">
            <name>ServerInfo</name>
        </reference>
        <reference name="LoginService">
            <name>JaasLoginService</name>
        </reference>
        <xml-reference name="LoginModuleConfiguration">
            <log:login-config xmlns:log="http://geronimo.apache.org/xml/ns/loginconfig-1.2">
                <log:login-module control-flag="REQUIRED" server-side="true" wrap-principals="false">
                    <log:login-domain-name>rollerdb</log:login-domain-name>
                    <log:login-module-class>org.apache.geronimo.security.realm.providers.SQLLoginModule</log:
login-module-class>
                    <log:option name="userSelect">SELECT username, passphrase FROM rolleruser WHERE username=?<
/log:option>
                    <log:option name="dataSourceApplication">null</log:option>
                    <log:option name="groupSelect">SELECT username, rolename FROM userrole WHERE username=?<
/log:option>
                    <log:option name="dataSourceName">rollerdb</log:option>
                </log:login-module>
            </log:login-config>
        </xml-reference>
    </gbean>
</web-app>
```

roller-custom.properties

```
#You may whant to set thous two properties
#uploads.dir=/var/local/roller/roller_data/uploads
#search.index.dir=/var/local/roller/roller_data/search-index

#---------------------------------
# Database configuration settings

# Hibernate dialect: You must override this to use a database other than MySQL4
#hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
hibernate.dialect=org.hibernate.dialect.DerbyDialect
```

Creating and Populating the Roller Derby Database

You will find a **create.db** file in the *roller/WEB-INF/dbscrips/* directory. This document will use and set up the **derby database** but you should be able to use any of the roller supported databases and as a alternative setup you will also find the mysql database pool below.

After starting Apache Geronimo log into the console and follow the given steps to create the **rollerdb** database.

1. In the console select the **DB Manager** link in the **Console Navigation** panel on the left.
2. Give the database name as **rollerdb** and click **Create** button.
3. Select **rollerdb** in the **Use DB** field.
4. Open **createdb.sql** in the **dbscrips/derby** directory from a text editor.
5. Paste the content **createdb.sql** into the **SQL Commands** text area and press **Run SQL** button.

Deploying Database Connection Pool Plan

The Roller application is going to access the **rollerdb** through a **connection pool**. Below you will find the steps to deploy the the roller database connection pool.

roller-derbydb-plan.xml

```
<?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>rollerdb</dep:artifactId>
            <dep:version>1.0</dep:version>
            <dep:type>rar</dep:type>
        </dep:moduleId>
        <dep:dependencies>
            <dep:dependency>
                <dep:groupId>org.apache.derby</dep:groupId>
                <dep:artifactId>derby</dep:artifactId>
                <dep:version>10.1.3.1</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>rollerdb</name>
                    <config-property-setting name="UserName">the_user</config-property-setting>
                    <config-property-setting name="Password">the_pw</config-property-
setting>
                    <config-property-setting name="Driver">org.apache.derby.jdbc.EmbeddedDriver</config-
property-setting>
                    <config-property-setting name="ConnectionURL">jdbc:derby:roller</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>
```

1. In the console Click on **Deploy New** link in the **Console Navigation** panel.
2. Load the **tranql-connector-ra-1.3.rar** into the **Archive input** box from the **<geronimo_home>/repository/org/tranql/tranql-connector-ra/1.3/tranql-connector-ra-1.3.rar** location.
3. Load the **roller-derbydb-plan.xml** into the **Plan** input box.
4. Press the **Install** button to deploy the connection pool into the server.

Building and deplpoy the Geronimo ready roller.war

Use the Java jar tool to reassemble your working area from the roller directory down naming it roller.war. Deploying Roller application is pretty straight forward as we are going to use the Geronimo Console.

1. Click the **Deploy New** link on the **Console Navigation** panel.
2. Load the **roller.war** file into the **Archive** input box.
3. Press **Install** button to deploy the roller application in the server.

Alternative database setup (MySql)

This is the MySql rollerdb plan just change the hibernate dialect setting in the roller-custom.properties re-pack the roller.war, create, build the database, deploy the plan and you will have roller up and running with the mysql database back end in minutes.

roller-mysqldb-plan.xml

```
<?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>rollerdb</dep:artifactId>
            <dep:version>1.0</dep:version>
            <dep:type>rar</dep:type>
        </dep:moduleId>
        <dep:dependencies>
            <dep:dependency>
                <dep:groupId>mysql</dep:groupId>
                <dep:artifactId>mysql-connector-java</dep:artifactId>
                <dep:version>3.1.12</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>rollerdb</name>
                    <config-property-setting name="UserName">the_user</config-property-setting>
                    <config-property-setting name="Password">the_pw</config-property-setting>
                    <config-property-setting name="Driver">com.mysql.jdbc.Driver</config-property-setting>
                    <config-property-setting name="ConnectionURL">jdbc:mysql://localhost:3306/roller</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>
```

Some problems and there solution

Hopefully this section eventually can be removed but at this writing there are a couple of things to look out for fortunately there are solutions. If you hit the following problem wile running Roller on Geronimo you should upgrade to a newer build of Geronimo 1.2 as this problem has been addressed by the developers.

Geronimo log message

```
--- ROOT CAUSE ---
java.lang.NullPointerException
       at org.apache.geronimo.connector.outbound.connectiontracking.ConnectionTrackingCoordinator.handleReleased
(ConnectionTrackingCoordinator.java:127)
       at org.apache.geronimo.connector.outbound.connectiontracking.
ConnectionTrackingCoordinator$$FastClassByCGLIB$$5d33aabf.invoke(<generated>)
       at net.sf.cglib.reflect.FastMethod.invoke(FastMethod.java:53)
```

If this error shows up in your Geronimo log wile starting Roller you are running in to a hibernate property loading problem. Hibernate is swalowing the activemq properties witout checking it first (this has been addressed in a newer versions of hibernate (3.2)).

Geronimo log message

```
13:13:34,849 FATAL [HibernateRollerImpl] Error initializing Hibernate
java.lang.ClassCastException: java.util.HashSet
        at org.hibernate.util.PropertiesHelper.resolvePlaceHolders(PropertiesHelper.java:88)
```

As Roller v3.1 is using hibernate v3.1 one (of at least 2) solution to this is to turn of activemq. This can be done by editing the /var/config/config.xml (make sure Geronimo is not running wile editing the file) and module load="false" the activemq-broker and activemq modules. If you need activemq running the solution should be to use a second set of properties for acitvemq (that dose not use a hash).