

JNDIDatasourceWithJBoss3.0.6

Switching from Cocoon managed datasource to JBoss JNDI datasource.

Up til now, I've used the datasource defined in cocoon which managed pooling, etc. The advantage of having jboss manage the connections is they can be shared across different applications running under jboss. Here is a list of things tha need to be done to make this work.

- move the jdbc driver jar file from the cocoon/WEB-INF/lib directory to jboss/server/default/lib. For Oracle, this file is ojdbc14.jar (This makes it available to all applications, plus jboss)
- create your own XXX-service.xml file. You can modify the one for your database found in jboss/docs/examples/jda. I took the oracle.service.xml and changed the "OracleDS" name and put in my own. Also edit the ConnectionURL, UserName, Password config properties to suit your connection.
- in the cocoon.xconf file, change your datasource from the <jdbc> style to the <j2ee> style. My example shows a datasource named "stevie" referring to the JNDI named source called "mbrdb".

```
<j2ee name="stevie">
  <dbname>mbrdb</dbname>
</j2ee>
```

- Here is a jboss-web.xml file that needs to go into your cocoon/WEB-INF directory with your web.xml file. just edit your datasource name, replacing "mbrdb" with your source name.

```
<?xml version="1.0"?>
<jboss-web>
  <resource-ref>
    <res-ref-name>jdbc/mbrdb</res-ref-name>
    <jndi-name>java:/mbrdb</jndi-name>
  </resource-ref>
</jboss-web>
```

- at the bottom of the web.xml file for your cocoon instance, add this;

```
<resource-ref>
  <res-ref-name>jdbc/mbrdb</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

It goes just before the </web-app> line at the bottom. You'll need to change the "mbrdb" reference here also.

- One other change I needed to make was to comment out 3 <map:action>s in the main sitemap.xmap file for cocoon. The first three actions defined are some "DatabaseXXXAction"s. They need to be commented out. Not sure why, but they caused cocoon to throw exceptions when used the <j2ee> datasource.

That is it! It took me many hours and the help of some co-workers to get this set up right. Now, in your cocoon pipeline, you can just use the datasource name like you did before the change!