

JBossDeployment

Warning: many pages on this wiki imply that a servlet container must be installed before running Cocoon, this is **wrong** - see [NoServletContainerRequired](#). A basic installation of Cocoon for test or development purposes can be *very simple*.

How to deploy Cocoon on JBoss and Tomcat

1. Deploy tomcat/jboss. Versions 4.0/2.4.3 are apparently known

to work. We use 4.0.4 and 2.4.4. "Note that according to the Cocoon2 homepage certain beta versions of Tomcat does not work with Cocoon2."

To deploy Tomcat 4.0.4 use the integrated 2.4.4/4.0.1 Tomcat/Jboss then copy a Tomcat 4.0.4 install over the "catalina" directory structure in the Jboss /Tomcat install. If you do this, I'd rename the base directory to help keep things straight as to what is installed.

2. Add environment variable CATALINA_OPTS=-Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=7070.[BR](#) This is to be able to debug java run in Tomcat. For Netbeans 3.2.1 for programming/debugging (remember to add port 7070 to debugging environment).

3. Delete in "[your path]/JBoss-2.4.3_Tomcat-4.0/jboss/lib" following:

- crimson.jar
- jaxp.jar
- xml.jar
- (xml.jar is not present in latest versions of JBoss.)

If Tomcat 3.x is used the following files must be also be deleted: from [your path]/JBoss-xxx_Tomcat-3.x/tomcat/lib:

- parser.jar
- jaxp.jar

4. new run.bat file in [your path]/JBoss-2.4.3_Tomcat-4.0/jboss/bin to:

```
@echo off
@if not "%ECHO%" == "" echo %ECHO%
@if "%OS%" == "Windows_NT" setlocal

set JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;run.jar

REM Add all login modules for JAAS-based security
REM and all libraries that are used by them here

REM need one of the two following lines for xerces support
set JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;../lib/xerces.jar
REM set JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;../lib/xml-apis.jar

REM Add the XML parser jars and set the JAXP factory names
REM Crimson parser JAXP setup(default)
REM set JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;../lib/crimson.jar
REM set JAXP=-Djavax.xml.parsers.DocumentBuilderFactory=org.apache.crimson.jaxp.DocumentBuilderFactoryImpl
REM set JAXP=%JAXP% -Djavax.xml.parsers.SAXParserFactory=org.apache.crimson.jaxp.SAXParserFactoryImpl
echo JBOSS_CLASSPATH=%JBOSS_CLASSPATH%
java %JAXP% -classpath "%JBOSS_CLASSPATH%" org.jboss.Main %1 %2 %3 %4 %5 %6 %7 %8 %9

pause
```

5. If you want to be able to debug JBoss/Tomcat/Cocoon2 replace the last line (before pause) with:

```
java -Xint -Xdebug -Xnoagent -classpath "%JBOSS_CLASSPATH%" -Xrunjdwp:transport=dt_socket,server=y,
address=12999,suspend=n org.jboss.Main %1 %2 %3 %4 %5 %6 %7 %8 %9
```

If you use Netbeans 3.2.1 for programming/debugging remember to add port 12999 to debugging environment.

6. copy xerces.jar to [yourpath]/JBoss-2.4.3_Tomcat-4.0/jboss/lib. We used versions 1.4.3. Download xerces binary and use .jar file in the downloaded xerces_xxx.zip The Apache site suggests that you should copy xml-apis.jar from cocoon/lib/core/ to jboss/lib. However, I did not have to do this. If you think you need xml-apis.jar you may also want to uncomment the line in run.bat that refers to it and comment the line that uses xerces. Only one of these should be needed...

7. add environment variable TOMCAT_HOME=[yourpath]/JBoss-2.4.3_Tomcat-4.0/catalina/ (if a Tomcat 3.x version is used "catalina" must be substituted with "tomcat"). This is for use with Cocoon ant file: build.bat.

8. Test that JBoss/Tomcat starts up and responds on port 8080/jboss Contrary to some documentation the test application still works fine.

Cocoon and Java 1.4 configuration

Cocoon requires more recent versions of the Xerces and Xalan libraries than those shipped with j2se 1.4. To override bundled libraries, follow these steps:

1. Create %JAVA_HOME%\jre\lib\endorsed directory.
2. Copy xerces-XXX.jar, xalan-XXX.jar, and the xml-apis.jar from the .\lib\core\ to the %JAVA_HOME%\jre\lib\endorsed\ directory.

Due to changes in JDBC between JDK 1.3 and JDK 1.4, it is not possible to use Cocoon built on JDK 1.3 with JDK 1.4 when it comes to database connections. Make sure you have a 1.4 compatible Cocoon build, or if you have the Cocoon source build it yourself with 1.4.

How to change Tomcat's Server port from 8080 to 80 using the JBoss-2.4.4-Tomcat-4.0.1 integrated package

JBoss will NOT read the server.xml of Catalina (tomcat4).

To change port, edit \$JBOSSHOME/conf/Catalina/jboss.jcml and look for the mbean configuration:

```
<mbean code="org.jboss.web.catalina.EmbeddedCatalinaServiceSX" name="DefaultDomain:service=EmbeddedTomcat" />
```

then change it as follows:

```
<mbean code="org.jboss.web.catalina.EmbeddedCatalinaServiceSX" name="DefaultDomain:service=EmbeddedTomcat">
  <attribute name="Port">80</attribute>
</mbean>
```

restart JBoss (with Tomcat/Catalina).

Instructions for Oracle 8.1.7 and JBoss 2.4.0

warning: see <http://marc.theaimsgroup.com/?l=xml-cocoon-users&m=103581609814762&w=2>

Using these instructions, you can use CMP beans or connect to Oracle via:

```
InitialContext context = new InitialContext();
DataSource ds = (DataSource) context.lookup("java:/ctdb");
```

1. Copy oracle's classes12.zip (this is available from Oracle technet, and has the jdbc driver) to \$JBOSS_DIST/lib/ext.
2. Note that jboss.properties does NOT need to be modified (as indicated in the jboss.properties file).
3. Modify \$JBOSS_DIST/conf/tomcat/jboss.jcml: Uncomment the XidClassName attribute shown below:

```
<!-- ===== -->
<!-- Transactions -->
<!-- ===== -->
<mbean code="org.jboss.tm.TransactionManagerService" name="DefaultDomain:service=TransactionManager">
  <attribute name="TransactionTimeout">300</attribute>

  <!-- Use this attribute if you need to use a specific Xid implementation -->
  <attribute name="XidClassName">oracle.jdbc.xa.OracleXid</attribute>
</mbean>
```

4. Also in jboss.jcml: add the Oracle driver to the JDBC drivers (this driver is in classes12.zip):

```

<!-- ===== -->
<!-- JDBC -->
<!-- ===== -->

<mbean code="org.jboss.jdbc.JdbcProvider" name="DefaultDomain:service=JdbcProvider">
  <attribute name="Drivers">org.hsqldb.jdbcDriver,org.enhydra.instantdb.jdbc.idbDriver,oracle.jdbc.driver.
OracleDriver</attribute>
</mbean>

```

5. Also in jboss.jcml in the JDBC section, define what the oracle datasource name is (say OracleDB), and the connection URL. In the code below, replace jdbc:oracle:thin:@<host>:<port>:<sid> with the appropriate string for your installation. Such as jdbc:oracle:thin:@192.168.1.111:1521:testdb

```

<mbean code="org.jboss.jdbc.XADataSourceLoader" name="DefaultDomain:service=XADataSource,name=datasourcename"
>
  <attribute name="PoolName">DefaultDS</attribute>
  <attribute name="DataSourceClass">org.jboss.pool.jdbc.xa.wrapper.XADataSourceImpl</a
ttribute>
  <attribute name="URL">jdbc:oracle:thin:@<host>:<port>:<sid></attribute>
  <attribute name="JDBCUser">username</attribute>
  <!-- set correct username here -->
  <attribute name="Password">password</attribute>
  <!-- set correct password here -->
</mbean>

```

or alternatively, this may be necessary:

```

<mbean code="org.jboss.jdbc.XADataSourceLoader" name="DefaultDomain:service=XADataSource,name=datasourcename"
>
  <attribute name="PoolName">poolname</attribute>
  <!-- as needed -->
  <attribute name="DataSourceClass">org.jboss.pool.jdbc.xa.wrapper.XADataSourceImpl</attribute>
  <attribute name="Properties"></attribute>
  <attribute name="URL">jdbc:oracle:thin:@sjmemctd2:1521:ctdev2</attribute>
  <attribute name="GCMinIdleTime">1200000</attribute>
  <attribute name="JDBCUser">username</attribute>
  <!-- set correct user name here -->
  <attribute name="MaxSize">10</attribute>
  <attribute name="Password">password</attribute>
  <!-- set correct password here -->
  <attribute name="GCEnabled">>false</attribute>
  <attribute name="InvalidateOnError">true</attribute>
  <attribute name="TimestampUsed">>false</attribute>
  <attribute name="Blocking">true</attribute>
  <!-- may cause exception if set false -->
  <attribute name="GCInterval">120000</attribute>
  <attribute name="IdleTimeout">1800000</attribute>
  <attribute name="IdleTimeoutEnabled">>false</attribute>
  <attribute name="LoggingEnabled">true</attribute>
  <attribute name="MaxIdleTimeoutPercent">1.0</attribute>
  <attribute name="MinSize">0</attribute>
</mbean>

```

6. Modify \$JBOSS_DIST/conf/tomcat/standardjaws.xml to use the proper datasource via the datasourceelement, and to use Oracle8 jdbc-sql type mappings. (This is for container managed persistence.) Make sure the datasource name matches whatever you defined for the datasource name in step 5.

```

<jaws>
  <datasource>java:/datasourcename</datasource>
  .
  .
  .

```

Authors:

written by Peter Hunsberger [original mail](#)

added to the docs by [ReinhardPoetz mail](#)