Installing Apache Cocoon (2.1 legacy document)

Table of contents

Copyright © 1999-2005 The Apache Software Foundation. All rights reserved.

Table of contents

Copyright © 1999-2005 The Apache Software Foundation. All rights reserved.

Warning:
This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

1. Getting Apache Cocoon

You have three choices for getting Cocoon: you can either download a stable release, or you can download development snapshot, or you can get the latest development version directly from the cvs repository.

1.1. Download a distribution

You can simply download the latest official release from the Cocoon distribution directory.

1.2. Download a development snapshot

You also can download one of the development snapshots from the CVS snapshots directory.

1.3. Step-by-step cvs instructions for Windows

See the Cocoon document Contrib for starting tips.

1. Download a recent WinCVS (1.2 or above);
2. Install it;
3. Start it;
4. Click on admin->preferences;
5. In "Enter the CVSROOT:" enter ":pserver:anoncvs@cvs.apache.org:/home/cvspublic" (without quotes);
6. In "Authentication:" choose "'passwd' file on the cvs server";
7. Click "Ok";
8. Click admin->login;
9. When asked for the password: answer "anoncvs" (without quotes);
10. Click "create->checkout module";
11. Module name and path on the server is "cocoon-2.1" (no quotes);
12. Choose a dir to put the source code in;
13. Go to the "Checkout-options" tab and select "By revision/tag/branch" and enter "HEAD";
14. Click "Ok";
15. If everything goes well, messages will start to appear in the log window;
16. Wait until you see "*****CVS exited normally with code 0*****" in the log window;
17. The Cocoon source is now on your harddrive.

1.4. Step-by-step cvs instructions for Unix

1. Start the shell of your choice.
2. Enter "cvs -d :pserver:anoncvs@cvs.apache.org:/home/cvspublic login".
3. When asked for the password: answer "anoncvs".
4. Enter "cvs -d :pserver:anoncvs@cvs.apache.org:/home/cvspublic -z3 checkout -r HEAD cocoon-2.1". This will create a directory called "cocoon-2.1" where the Cocoon2 source will be stored.
5. Wait until cvs has finished.
6. The Cocoon source is now on your harddrive.

In case you want to update your Cocoon source tree to the current version, change to the "cocoon-2.1"

Copyright © 1999-2005 The Apache Software Foundation. All rights reserved.

directory and call "cvs -z3 update -d -P".

2. Configuring Environment

2.1. Set JAVA_HOME environment variable

Set the JAVA_HOME environment variable to point to the root directory of the Java Development Kit installed on your machine. To do this simply type:

[unix] JAVA_HOME=/path/to/java/ [win32] SET JAVA_HOME=c:\path\to\java

Your mileage may vary, but you know how to setup environments, right?

2.2. 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. (Tomcat users use %TOMCAT_HOME\common\endorsed instead)
2. Copy xercesXXX.jar, xalan-XXX.jar, and the xml-apis.jar from the .\lib\endorsed\ to the new directory created above.

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 prepare cocoon with a JDK that matches the one you will deploy on.

2.3. JDK Dependency

Cocoon requires a Java compiler for installation and for running some components like XSP etc. For components, the default configuration of Cocoon does not use the compiler in JAVA_HOME, but a version shipped with Cocoon.

You can configure which compiler Cocoon uses in WEB-INF/lib.ghoward: Is this still an issue?

2.4. UNIX with X server

Cocoon is bundled with the Batik (SVG rasterization toolkit) to deliver SVG imaging capabilities. Batik uses Java java.awt library, which (at least in Sun JDK before 1.4) requires graphics display. This means that X server must be running and Cocoon should have permission to connect to it.

Easiest way to avoid X server connection problem (and to have mentioned permission) is to install and run Cocoon and entire servlet engine of your choice under regular user account.

For security, and many other reasons, X server can be replaced by Xfvb or PJA (alternative awt implementation).

Sun JDK 1.4 does not require graphics display anymore, but Java has to be started with the argument -Djava.awt.headless=true, and X libraries still must be installed.ghoward: Is this still an issue?

2.5. Headless UNIX and PJA

If you are using unix with the Sun JDK 1.4, it can run in the headless environment (but you still must have X libraries installed!) when following option is provided on Java startup:

Copyright © 1999-2005 The Apache Software Foundation. All rights reserved.

-Djava.awt.headless=trueIf you use Tomcat, this can be done by setting environment variable CATALINA_OPTS (Tomcat 4.x), or TOMCAT_OPTS (Tomcat 3.x):export CATALINA_OPTS='-Djava.awt.headless=true'

If you are using unix with the Sun JDK 1.3.1 or earlier, it's awt implementation requires you to use X even if you aren't actually displaying anything. One simple solution is to use a different implementation of the awt.

1. From www.eteks.com you can get an awt replacement that doesn't need X: http://www.eteks.com/pja/en/.
2. Then add the following options to the Java command starting your container:

-Xbootclasspath/a:/path/to/pja.jar -Dawt.toolkit=com.eteks.awt.PJAToolkit -Djava.awt.graphicsenv=com.eteks.java2d.PJAGraphicsEnvironment -Djava.awt.fonts=/usr/local/jdk/jre/lib/fonts/ If you use Tomcat, this can be done by setting environment variable CATALINA_OPTS (Tomcat 4.x), or TOMCAT_OPTS (Tomcat 3.x):export CATALINA_OPTS='-Xbootclasspath/a:/path/to/pja.jar \
-Dawt.toolkit=com.eteks.awt.PJAToolkit \
-Djava.awt.graphicsenv=com.eteks.java2d.PJAGraphicsEnvironment \
-Djava.awt.fonts=/usr/local/jdk/jre/lib/fonts/'

3. Building Cocoon

3.1. Optional functionality

This is an area that has changed significantly since 2.0

3.1.1. Blocks

Additional "components" in Cocoon 2.1 are now implemented using a partial implementation of a concept called "blocks". Full support (planned for the next release) will include hot-deployable services with java-like extension and inheritance. For this release, blocks are implemented as self contained units of code, samples, files and configuration. Most, if not all optional features have been factored out into blocks and can be neatly included or excluded as a unit.

Most blocks are configured by default to be included in the build, but can be excluded using a local.blocks.properties file.

Some blocks delivered with Cocoon require additional libraries which can not be redistributed with Cocoon (e.g. the Php block). The documentation for these should provide more information, and you should find a "mocks" directory containing non-functional code provided merely to allow compilation.

3.1.2. Optional Jars

Some additional libraries could not be factored ou into a block. For example, the Jakarta Commons HttpClient jar is used by several different optional components, some not in blocks. If you determine that one of these is not necessary for your build, you can eliminate it from your build by removing its entry in lib/jars.xml and removing the jar from lib/optional.

3.1.3. Building a minimal Cocoon

By creating and editing local.build.properties and local.blocks.properties you can remove any unnecessary features you desire.

3.2. Running the build

Copyright © 1999-2005 The Apache Software Foundation. All rights reserved.

This is an area that has changed significantly since 2.0

Cocoon uses Jakarta Ant for the build and installation, and comes with a build script ([unix]./build.sh, [win32] .\build.bat) that automates the process.

If you want to use build.xml directly with your copy of Ant, please run the build script at least once after every CVS checkout, to ensure that extra initializations like jar copying are done correctly.The build script overrides the existing ANT_HOME variable.

There are basically two options that can be set as parameters to the script: **targets** and **properties**.

The use of -D style properties as parameters to the build is deprecated. Use the build and blocks properties files instead.

3.2.1. About build targets

Targets are the execution units available in Ant build files.

Targets are specified by appending them to the script invocation:

[unix] ./build.sh target1 target2 ... [win32] .\build.bat target1 target2 ...

3.2.2. Cocoon build targets

The **build.xml** file comes with some basic important targets. If no target is specified, the default one is used (defined in build.xml).

Cocoon targets place work files and results in a build directory under the cocoon root. The only exception are the distribution targets that build in a directory called dist.

The key targets you will use most often are noted below. For a listing of all available targets, run:

[unix] ./build.sh -projecthelp [win32] .\build.bat -projecthelp

3.2.2.1. build clean

Cleans the build directory. It is recommended to clean Cocoon build directory every time you upgrade Cocoon, or add/remove libraries from the ./lib/optional/ directory, or change JDK version.

3.2.2.2. build webapp [default]

Builds a Cocoon web application in build/webapp. This is the default target, so running the build script with no arguments calls this target.

For quick testing, a stripped down version of Jetty is included in the distribution. After a build webapp Jetty can be launched by cocoon servlet and Cocoon will be available at http://localhost:8888/.

3.2.2.3. build war

Builds a Cocoon web application in build/webapp and packages it as a .war file.

4. Installing Cocoon

In most servlet engines, this is just a matter of copying the war file or webapp directory to a specific directory and the engine will take care of installing it when restarted.

If you are using JDK 1.4, be sure you have read and followed the "Java 1.4 configuration" information above.

Copyright © 1999-2005 The Apache Software Foundation. All rights reserved.

If those simple instructions do not work as expected, you may find help in some of the container-specific notes below contributed by users:

4.1. Installing on Tomcat 3.3.X

This is a very easy installation.

1. Build the Cocoon webapp as described above.
2. Copy cocoon/build/cocoon/cocoon.war into tomcat/webapps directory.
3. Start Tomcat: Go to the tomcat/bin directory, and run the startup script.
4. Open the Cocoon welcome page: http://localhost:8080/cocoon/
5. Congratulations! You should see the Cocoon welcome page.

4.2. Installing on Tomcat 3.2.X

Cocoon requires Tomcat version 3.2 or greater. It wouldn't work with Tomcat 3.1.X

Tomcat currently uses a different version of the XML parser than Cocoon. To get Cocoon to work, you need to perform the following steps:

1. **Stop Tomcat** Go to the tomcat/bin directory, and run the shutdown script.
2. **Delete tomcat/lib/jaxp.jar** Tomcat's jaxp.jar is 'sealed', and since xerces contains its own implementation of the JAXP standard extension, Java will fail to load xerces and report a 'Package Sealing Violation' if both are in the classpath.
3. **Rename tomcat/lib/parser.jar to tomcat/lib/zparser.jar** Tomcat's parser.jar contains older versions of some the same XML APIS that Xerces uses, and these will prevent Xerces from functioning properly if they appear before Xerces in the classpath. Since Tomcat's startup scripts automatically load all the jar files in tomcat/lib in name order, changing the name of the file causes it to be loaded last in the classpath.
4. **Copy the cocoon/lib/core/xerces-XXX.jar and cocoon/lib/core/xml-apis.jar JAR files to tomcat/lib** Cocoon will now be able to see and use the correct XML libraries.
5. **Copy cocoon/build/cocoon/cocoon.war into tomcat/webapps**
6. **Start Tomcat** Go to the tomcat/bin directory, and run the startup script.
7. **Start using Cocoon** Access the URI http://localhost:8080/cocoon/ with your favorite browser and start to enjoy the world of Cocoon.

4.3. Installing on Tomcat 4.0 - 4.0.1, 4.0.4b1

Tomcat 4 is a really straight-forward installation.

1. Build the Cocoon webapp as described above.
2. Copy cocoon/build/cocoon/cocoon.war into tomcat/webapps directory.
3. Start Tomcat: Go to the tomcat/bin directory, and run the startup script.
4. Open the Cocoon welcome page: http://localhost:8080/cocoon/
5. Congratulations! You should see the Cocoon welcome page.

4.4. Installing on Tomcat 4.0.3

If you have to use Tomcat 4.0.3, you have to replace its XML parser with the one shipped with Cocoon.

1. Remove tomcat/common/lib/xerces.jar file.
2. Copy following libraries from the cocoon/lib/core directory to the tomcat/common/lib directory:
   - xalan-XXX.jar
   - xercesImpl-XXX.jar
   - xml-apis.jar

Copyright © 1999-2005 The Apache Software Foundation. All rights reserved.

3. Copy cocoon/lib/optional/batik-all-XXX.jar to the tomcat/common/lib directory.
4. Edit extra-classpath parameter in the cocoon/src/webapp/WEB-INF/web.xml file:

For UNIX:

<init-param> <param-name>extra-classpath</param-name>
<param-value>/tomcat/common/lib/xalan-XXX.jar: /tomcat/common/lib/xercesImpl-XXX.jar:
/tomcat/common/lib/xml-apis.jar: /tomcat/common/lib/batik-all-XXX.jar</param-value>
</init-param>

For Windows:

<init-param> <param-name>extra-classpath</param-name>
<param-value>C:\tomcat\common\lib\xalan-XXX.jar;
C:\tomcat\common\lib\xercesImpl-XXX.jar; C:\tomcat\common\lib\xml-apis.jar;
C:\tomcat\common\lib\batik-all-XXX.jar</param-value> </init-param> param-value should be in one line! Also, replace /tomcat/ (UNIX), C:\tomcat\ (Windows) with the path to your Tomcat installation home.

1. Clean Cocoon build directory: build clean
2. Build Cocoon webapp: build webapp
3. Remove xalan-XXX.jar, xercesImpl-XXX.jar, batik-all-XXX.jar, and xml-apis.jar from the cocoon.war archive.
4. Copy cocoon/build/cocoon/cocoon.war into tomcat/webapps directory.
5. Start Tomcat: Go to the tomcat/bin directory, and run the startup script.
6. Open the Cocoon welcome page: http://localhost:8080/cocoon/
7. Congratulations! You should see the Cocoon welcome page.

4.5. Installing on Tomcat 4.0.4b1 LE with JDK 1.4.0

This combination is also easy to install.

1. Build the Cocoon webapp as described above.
2. Copy cocoon-2.1/build/cocoon/cocoon.war into tomcat/webapps directory.
3. Set environment variable CATALINA_OPTS=-Djava.awt.headless=true
4. Start Tomcat: Go to the tomcat/bin directory, and run the startup script.
5. Open the Cocoon welcome page: http://localhost:8080/cocoon/
6. Congratulations! You should see the Cocoon welcome page.

Make sure that JAVA_HOME enviroment variable points to the JDK1.4.0. If you had JDK1.3.1 or earlier before, build clean before all these steps.

4.6. Installing on BEA Weblogic 6.0sp2

This installs Cocoon using the cocoon.war file. This was successfully installed under Windows 2000 and JDK 1.3.1. Unix users will need to adjust appropriately. If you haven't done so already, build a domain and a server. In this discussion, the name of the domain is 'mydomain', the name of the server is 'myserver', and WebLogic installation directory is c:\bea\wlserver6.0sp2\. These are the BEA defaults.

1. Build the Cocoon webapp as described above.
2. Copy cocoon\build\cocoon\webapp directory into the c:\bea\wlserver6.0sp2\config\mydomain\applications\ directory of your WebLogic server.
3. Copy the xerces-XXX.jar and xml-apis.jar JAR files from the cocoon\lib\core\ to the directory of your choice, say c:\bea\.
4. Add to the config.xml of the WebLogic server following snippet:

<Application Deployed="true" Name="Cocoon" Path="./config/mydomain/applications">

Copyright © 1999-2005 The Apache Software Foundation. All rights reserved.

<WebAppComponent Name="cocoon" Targets="myserver" URI="cocoon"/> </Application>

1. Edit c:\bea\wlserver6.0sp2\config\mydomain\startWebLogic.cmd file, add xerces and xml-apis JAR files to the classpath:

set CLASSPATH=c:\bea\xerces-XXX.jar;c:\bea\xml-apis.jar set CLASSPATH=%CLASSPATH%;.;.\lib\weblogic_sp.jar set CLASSPATH=%CLASSPATH%;.\lib\weblogic.jar

1. Start WebLogic server using startWebLogic.cmd.
2. Using a browser, you might want to check WebLogic configuration using console: http://localhost:7001/console/.
3. Open the Cocoon welcome page: http://localhost:7001/cocoon/ (Don't forget the final '/' in the link.)
4. Congratulations! You should see the Cocoon welcome page.

Because of some issues with this version of WebLogic, you will see lots of exceptions in the WebLogic's console window.

4.7. Installing on ServletExec 3.1 (In Process with IIS)

This installs Cocoon in a "war" configuration. This was successfully installed under Windows NT 4.0 and IIS 4. I don't believe that SE is available for unix.

Please note that *JDK 1.3* is required.

1. Install IIS as usual
2. Install ServletExec (default paths will be used throughout), but don't start it.
3. Build Cocoon's war file (include lib's)
4. Copy *cocoon.war* into *C:\Program Files\New Atlanta\ServletExec ISAPI\webapps\default*, creating the directory default if required.
5. Start IIS.
6. Open the Cocoon welcome page (http://localhost/cocoon/)
7. Congratulations! (hopefully) you should see the Cocoon welcome page.

4.8. Installing on JBoss 2.4.4 with Tomcat 4.0.1 (Catalina)

This section describes the deployment of the Cocoon sample WAR with the JBoss-2.4.4_Tomcat-4.0.1 package. It assumes that you built Cocoon as described above or downloaded the binary Cocoon distribution. All steps have been tested with a fresh JBoss 2.4.4 installation on Linux and Windows 2000.

The JBoss/Tomcat bundle is available from the JBoss project page

The JBoss/Tomcat package has the following directory structure

[path]/JBoss-2.4.4_Tomcat-4.0.1/jboss [path]/JBoss-2.4.4_Tomcat-4.0.1/catalina

Subsequently,

- jboss denotes the JBoss-2.4.4_Tomcat-4.0.1/jboss directory
- catalina is short for JBoss-2.4.4_Tomcat-4.0.1/catalina
- and cocoon is the base directory of your Cocoon distribution or CVS checkout.

In order to get Cocoon running you have to install Xerces as default XML parser for JBoss.

- Stop JBoss if it is running.
- Remove the following files from the jboss/lib directory
  - crimson.jar
  - jaxp.jar
- Copy xml-apis.jar from cocoon/lib/core/ to jboss/lib

Copyright © 1999-2005 The Apache Software Foundation. All rights reserved.

- Change jboss/bin/run.sh

[...] # Add the XML parser jar and set the JAXP factory names # Crimson parser JAXP setup(default) **# Change it to Xerces for C2** JBOSS_CLASSPATH=$JBOSS_CLASSPATH:**../lib/xml-apis.jar # Remove the following two lines** JAXP=-Djavax.xml.parsers.DocumentBuilderFactory=\ org.apache.crimson.jaxp.DocumentBuilderFactoryImpl JAXP="$JAXP -Djavax.xml.parsers.SAXParserFactory=\ org.apache.crimson.jaxp.SAXParserFactoryImpl" [...] Windows users have to change run.bat accordingly.

- Start JBoss with run_with_catalina.sh or run_with_catalina.bat
- Copy cocoon/build/cocoon/cocoon.war to jboss/deploy
- Check the server log to make sure that J2EE application: [...]/cocoon.war is deployed.
- Open the Cocoon welcome page (http://localhost:8080/cocoon/)
- You should see the Cocoon welcome page.

As both JBoss and Cocoon ship with a Hypersonic database installed, these two conflict and you won't be able to use Cocoon database (SQL) samples. Then again, you probably use JBoss for EJB persistence anyway, so this shouldn't bother you too much ;-)

4.9. Installing on JBoss 2.2.2 with Tomcat 3.2.2

This section describes the deployment of the Cocoon sample WAR with the JBoss 2.2.2/Tomcat-3.2.2 package. It assumes that you built Cocoon as described above. All steps have been tested with a fresh JBoss 2.2.2 installation on Linux and Windows ME(sic).

The JBoss/Tomcat bundle is available from the JBoss project page

The JBoss/Tomcat package has the following directory structure

[path]/JBoss-2.2.2_Tomcat-3.2.2/jboss [path]/JBoss-2.2.2_Tomcat-3.2.2/tomcat

Subsequently,

- jboss denotes the JBoss-2.2.2_Tomcat-3.2.2/jboss directory
- Tomcat is short for JBoss-2.2.2_Tomcat-3.2.2/tomcat
- and cocoon is the base directory of your Cocoon distribution or CVS checkout.

In order to get Cocoon running you have to install Xerces as default XML parser for JBoss.

- Stop the server if it is running.
- Remove the following files from the jboss/lib directory
  - crimson.jar
  - jaxp.jar
  - xml.jar
- Remove the following files from the tomcat/lib directory
  - jaxp.jar
  - parser.jar
- Copy xerces-XXX.jar from cocoon/lib/core/ to jboss/lib
- Change jboss/bin/run.sh

[...] # Add the XML parser jars and set the JAXP factory names # Crimson parser JAXP setup(default) **# Change it to Xerces for C2** JBOSS_CLASSPATH=$JBOSS_CLASSPATH:**../lib/xerces-XXX.jar # Remove the following two lines** JAXP=-Djavax.xml.parsers.DocumentBuilderFactory=\ org.apache.crimson.jaxp.DocumentBuilderFactoryImpl JAXP="$JAXP -Djavax.xml.parsers.SAXParserFactory=\ org.apache.crimson.jaxp.SAXParserFactoryImpl"

Copyright © 1999-2005 The Apache Software Foundation. All rights reserved.

[...] Windows users have to change run.bat accordingly.
- Start JBoss with run_with_tomcat.sh or run_with_tomcat.bat
- Copy cocoon/build/cocoon/cocoon.war to jboss/deploy
- Check the server log to make sure that J2EE application: [...]/cocoon.war is deployed.
- Open the Cocoon welcome page (http://localhost:8080/cocoon/)
- Congratulations! (hopefully) you should see the Cocoon welcome page.

As both JBoss and Cocoon ship with a Hypersonic database installed, these two conflict and you won't be able to use Cocoon database (SQL) samples. Then again, you probably use JBoss for EJB persistence anyway, so this shouldn't bother you too much ;-)

4.10. Installing on Resin 2.x

This section describes the deployment of the Cocoon sample WAR with Resin 2.x. It assumes that you built Cocoon as described above. All steps have been tested with a fresh Resin 2.0.0, 2.0.4, and 2.1.3 installations (the package is available from Resin's download page)

After unpacking the Resin package you get the following directory structure

[path]... [path]/resin-2.x/conf [path]/resin-2.x/lib [path]/resin-2.x/webapps [path]...

To get Cocoon running do the following:

- Stop the server if it is running.
- **For 2.0.3 version and older:** If yours Resin is older then 2.0.4, you have to install Xerces as default XML parser for Resin
  - Remove the following files from the resin-2.0.x/lib directory:
    - jaxp.jar
    - dom.jar
    - sax.jar
  - Copy xerces-XXX.jar and xml-apis.jar JAR file from cocoon-2.1/lib/core/ to the resin-2.0.x/lib/ directory.
- **For 2.0.4 version and newer:** Edit resin-2.x/conf/resin.conf, change value of the servlet-classloader-hack element to true
- Copy the cocoon-2.1/build/cocoon/cocoon.war WAR file to resin-2.x/webapps directory
- Start Resin as usual
- Open the Cocoon welcome page (http://localhost:8080/cocoon/)
- Congratulations! (hopefully) you should see the Cocoon welcome page.

If you want to place Cocoon webapp in a directory different than resin-2.x/webapps, you need to edit resin-2.x/conf/resin.conf file and add a line somewhere in <host> tag: <web-app id='/cocoon' app-dir='/path/to/webapp/cocoon.war'/>

4.11. Installing on HP-AS 8.X

HP-AS is J2EE application server available from the Hewlett-Packard website. Download and install HP-AS 8.X

Cocoon cannot be deployed as a .war file in HP-AS. Use the following steps to deploy cocoon.war:
1. Extract the cocoon.war file to some directory, using WinZIP or a similar utility to extract the files.
2. To run HP-AS, go to **(Start | Programs | HP Middleware | HP Application Server | System Console).**
   The HP-AS Console appears with a Log browser. As the HP-AS kernel starts and initializes, messages will appear in the status bar of the console. Wait for the message 'Kernel started' to

Copyright © 1999-2005 The Apache Software Foundation. All rights reserved.

appear in the Log browser. The following message should display:
[10/16/01 16:03:50][localhost_][S]:Kernel "kernel" started.

3. To verify that an instance of HP-AS is running, open a web browser and go to http://localhost:9090/helloservlet/hello
   An HTML page should appear containing the following message:
   Congratulations!
   Congratulations from the HelloWorldServlet
   It appears you have the server running
   My servlet path is /hello
   This test is valid only if you've performed a full install of HP-AS
4. In the HP-AS console, select **View | Deployment Window**. In the **Available Files** pane on the right, browse to the the directory you extracted the cocoon.war file to.
   Expand this directory, and then drag and drop the cocoon sub-directory node to the **kernel.j2ee-partition** icon in the left pane.
5. When prompted, answer **Yes** to the deployment question.
   This should create an appropriate entry in the HP-AS j2ee partition configuration file.
   In the current version of the console, there is no indication that the operation succeeded. If you see a parser error in the Log browser, ignore it.
6. To test the deployment, open a web browser and go to the following URL:
   http://localhost:9090/cocoon/welcome
   Congratulations! (hopefully) you should see the Cocoon welcome page. (this request may take some time).

4.12. Installing on JRun 3.1

This section describes the deployment of the Cocoon sample WAR with JRun 3.1, on its default server. It assumes that you built Cocoon as described above. All steps have been tested under Win2000.

To get Cocoon running do the following:

- Stop the default and admin servers if they are running.
- Remove jaxp.jar and parser.jar files (Crimson XML parser) from the jrun/lib/ext/ directory.
- Install Xerces as default XML parser for JRun by copying xerces-XXX.jar and xml-apis.jar JAR files from the cocoon-2.1/lib/core/ to jrun/lib/ext/ directory.
- Update Rhino shipped with JRun with newer version from the Cocoon by overwriting jrun/lib/rhino.jar JAR file with the cocoon-2.1/lib/optional/rhino-1.5r3.jar file.
- Start JRun admin server.
- Start JRun default server.
- Open JRun admin page: http://localhost:8000/
- Deploy cocoon.war webapp using console. Use same values for application name and URI prefix (e.g., application name "cocoon", URI "/cocoon").
- Open the Cocoon welcome page: http://localhost:8100/cocoon/
- Congratulations! (hopefully) you should see the Cocoon welcome page.

Instead of deploying WAR file using console, same could be done by copying cocoon-2.1/build/cocoon/webapp under jrun/servers/default/ directory and adding following lines to the jrun/servers/default/local.properties:cocoon.rootdir=/absolute/path/to/jrun/servers/default/cocoon cocoon.class={webapp.service-class} webapp.mapping./cocoon=cocoon

4.13. Installing on iPlanet Web Server 4.x and other engines without context management

Copyright © 1999-2005 The Apache Software Foundation. All rights reserved.

iPlanet Web Server 4.x provides the servlet 2.2 API (javax.servlet.* classes), but the servlet engine doesn't handle servlet contexts. This means there is no classloader built with the contents of WEB-INF/classes and WEB-INF/lib and that resolution of context resources (using ServletContext.getResource()) doesn't give the expected results.

To be able to run on such non-compliant engines, Cocoon provides a bootstrap servlet in org.apache.cocoon.BootstrapServletthat handles all the servlet context related behaviours needed for proper functioning.

To use this bootstrap servlet, configure your servlet engine as follows (how to do it depends on the actual engine - see below for iPlanet) :

- add cocoon.jar (and only this one) in the engine's classpath,
- declare the org.apache.cocoon.servlet.BootstrapServlet servlet,
- add a "context-directory" parameter, whose value is the absolute path to Cocoon's context (e.g. "/path/webapp/cocoon"),
- add any other cocoon parameters you want to this servlet (see web.xml for a description of available parameters),
- configure a path translation from "/" to the servlet.

For iPlanet Web Server 4.x, this translates to :

- connect to the administration server of your web server,
- in the "Servlet" tabs, select "Configure servlet attributes", and enter the following :
  - Servlet Name : cocoon
  - Servlet Code (class name) : org.apache.cocoon.servlet.BootstrapServlet
  - Servlet Classpath : /path/webapp/cocoon/WEB-INF/lib/cocoon.jar
  - Servlet Args : context-directory=/path/webapp/cocoon (and any other Cocoon parameters you want)
- select "Configure Servlet Virtual Path Translation" and enter the following :
  - Virtual Path : @/.*
  - Servlet Name : cocoon
- save and apply your changes, and enjoy the latest Cocoon on an old-fashioned servlet engine!

4.14. Installing on WebSphere 4.0

This section describes the deployment of the Cocoon sample WAR with WebSphere 4.0, on its default server. It assumes that you built Cocoon as described above. All steps have been tested under Win2000 and WebSphere AEs 4.0.1 a0136.02.

To get Cocoon running do the following:

- Start the server using startServer startup script.
- Open admin page: http://localhost:9090/admin/
- Deploy cocoon.war webapp using console.
- Save server configuration file.
- Restart the server using stopServer and startServer scripts.
- Open the Cocoon welcome page: http://localhost:9080/cocoon/
- Congratulations! (hopefully) you should see the Cocoon welcome page.

WebSphere power users might deploy Cocoon by exploding cocoon.war into installedApps directory and editing config/server-cfg.xml file.

Copyright © 1999-2005 The Apache Software Foundation. All rights reserved.

1. Comments

add your comments

Copyright © 1999-2005 The Apache Software Foundation. All rights reserved.