

HotDeploy

Based on mail of Konstantin Piroumian at cocoon-users@xml.apache.org.

Originally by Vadim Gritsenko's suggestions. – KP

To avoid restarting the VM (the servlet container) you should do the following:

Use a container that supports hot deploy or so (I've used Tomcat 4.0.4 and 4.1). You can use it for development only and then deploy your application to a production server.

Configure your servlet container to load application located in

```
<cocoon-home>/build/cocoon/webapp
```

In case of Tomcat you should add a context in

```
<tomcat-home>/conf/server.xml
```

pointing to that location, e.g.:

```
<Context path="/cocoon" docBase="C:\xml-cocoon2\build\cocoon\webapp"
  debug="0" reloadable="true">
  <Logger className="org.apache.catalina.logger.FileLogger"
    prefix="localhost_cocoon_log." timestamp="true" suffix=".txt"/>
  <Resources className="org.apache.naming.resources.FileDirContext" allowLinking="true"/>
</Context>
```

comments:

- <Logger/> is for creating another log file for Cocoon specific log messages.
- <Resources/> is only needed, if your webapp directory contains symbolic links or the docbase itself is a symbolic link.

At last build Cocoon using

```
build.[bat|sh] -Dinclude.webapp.libs=true webapp-local
```

This will build Cocoon and will place all classes in WEB-INF/classes so Tomcat could track changes.

Now you can compile your classes using

```
build.[bat|sh] webapp-local
```

and Tomcat will reload the application as soon as modifications are detected (it requires about a 1 minute).

important note:

Your webapp must not be deployed within the webapps Directory if you use this approach. Otherwise your cocoon app would be started twice!