

EXistInCocoon

Integrating eXist into an existing Cocoon 2.0.4 installation

eXist comes with its own pre-installed Cocoon. However, integrating eXist into an already running Cocoon is not very difficult. Follow the steps below to integrate eXist 0.9 into an existing Cocoon 2.0.4 web application.

The following paragraphs assume that you have already set up a web application based on Cocoon.

First create a directory named data in the WEB-INF folder of your Cocoon based web application. This is where the database files will be created by eXist.

Next, copy the conf.xml configuration file from eXist into the WEB-INF folder of your Cocoon application. Additionally, you have to copy a number of jar-files from eXist's WEB-INF/lib folder into the WEB-INF/lib folder of your Cocoon application:

- exist.jar
- antlr.jar
- axis.jar
- clutil.jar
- commons-collections-2.1.jar
- commons-discovery.jar
- commons-httpclient-2.0.2.jar
- commons-jxpath-1.0.jar
- commons-logging-1.0.jar
- jakarta-oro-2.0.6.jar
- jaxrpc.jar
- libreadline-java.jar
- log4j.jar
- saaj.jar
- trove.jar
- tt-bytecode.jar
- wsdl4j.jar
- xmldb.jar
- xmlrpc-1.1.jar

If you want WebDAV access you'll also need:

- jakarta-tomcat-4.0.3/webapps/exist/WEB-INF/lib/dom4j.jar
- jakarta-tomcat-4.0.3/webapps/exist/WEB-INF/lib/xincon.jar
- jakarta-tomcat-4.0.3/webapps/exist/WEB-INF/lib/catalina-util.jar

Next step: edit the web.xml file in your WEB-INF directory. From the web.xml file that comes with eXist copy the servlet definitions for the RpcServlet, DatabaseAdminServlet, AxisServlet, AdminServlet and (optionally) the xincon servlet into your own web.xml. Additionally, copy all the servlet-mapping sections for these servlets. Please note that order is important in the web.xml file: the block with <servlet> sections should always precede the <servlet-mapping> sections.

An example web.xml file is shown below:

Example: web.xml example (Cocoon CVS version 2.1 and eXist CVS version)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
This is the web-app configurations that allow Cocoon to work under
Apache Tomcat. Please, follow the installation section of the
documentation for more information about installing Cocoon on Tomcat
-->
<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
<web-app>
  <display-name>eXist Server</display-name>
  <description>eXist Server Setup</description>
  <!-- Servlets required by eXist -->
  <!-- RpcServlet provides XML-RPC access to eXist -->
  <servlet>
    <servlet-name>org.exist.xmlrpc.RpcServlet</servlet-name>
    <servlet-class>org.exist.xmlrpc.RpcServlet</servlet-class>
  </servlet>
  <!-- DatabaseAdminServlet: used to start/stop the database. -->
  <servlet>
    <servlet-name>org.exist.DatabaseAdminServlet</servlet-name>
    <servlet-class>org.exist.DatabaseAdminServlet</servlet-class>
  <!-- where to find eXist's configuration file
relative to basedir
```

```

-->
    <init-param>
        <param-name>configuration</param-name>
        <param-value>conf.xml</param-value>
    </init-param>
<!-- eXist's home directory. All file names in
the configuration file will be relative to this
directory.
-->
    <init-param>
        <param-name>basedir</param-name>
        <param-value>WEB-INF</param-value>
    </init-param>
    <init-param>
        <param-name>start</param-name>
        <param-value>true</param-value>
    </init-param>
    <load-on-startup>2</load-on-startup>
</servlet>
<!-- configure the Axis servlet. Axis provides eXist's
web-services via SOAP -->
    <servlet>
        <servlet-name>AxisServlet</servlet-name>
        <display-name>Apache-Axis Servlet</display-name>
        <servlet-class>org.apache.axis.transport.http.AxisServlet</servlet-class>
    </servlet>
    <servlet>
        <servlet-name>AdminServlet</servlet-name>
        <display-name>Axis Admin Servlet</display-name>
        <servlet-class>org.apache.axis.transport.http.AdminServlet</servlet-class>
        <load-on-startup>100</load-on-startup>
    </servlet>
<!-- xincon: Provides Webdav for eXist -->
    <servlet>
        <servlet-name>xincon</servlet-name>
        <servlet-class>xincon.WebdavServlet</servlet-class>
        <init-param>
            <param-name>dbroot</param-name>
            <param-value>xmldb:exist:///db</param-value>
        </init-param>
<!-- log4j parameters -->
    <init-param>
        <param-name>log4j-priority</param-name>
        <param-value>DEBUG</param-value>
    </init-param>
    <init-param>
        <param-name>log4j-layout</param-name>
        <param-value>%-5p [%M] %m%n</param-value>
    </init-param>
<!-- end log4j parameters -->
<!-- load this servlet first to avoid conflicts with
eXist's log-configuration -->
    <load-on-startup>1</load-on-startup>
</servlet>
<!-- Configure Cocoon2 -->
    <servlet>
        <servlet-name>Cocoon2</servlet-name>
        <display-name>Cocoon2</display-name>
        <description>The main Cocoon2 servlet</description>
<!-- I'm skipping Cocoon configuration here ... -->
    </servlet>

    <servlet-mapping>
        <servlet-name>org.exist.xmlrpc.RpcServlet</servlet-name>
        <url-pattern>/xmlrpc</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>org.exist.DatabaseAdminServlet</servlet-name>
        <url-pattern>/admin</url-pattern>
    </servlet-mapping>
    <servlet-mapping>

```

```

    <servlet-name>xincon</servlet-name>
    <url-pattern>/webdav/*</url-pattern>
  </servlet-mapping>
<!-- Axis servlets -->
  <servlet-mapping>
    <servlet-name>AxisServlet</servlet-name>
    <url-pattern>servlet/AxisServlet</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>AxisServlet</servlet-name>
    <url-pattern>*.jws</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>AxisServlet</servlet-name>
    <url-pattern>/services/*</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>AdminServlet</servlet-name>
    <url-pattern>servlet/AdminServlet</url-pattern>
  </servlet-mapping>
<!--
Cocoon handles all the URL space assigned to the webapp using its sitemap.
It is recommended to leave it unchanged. Under some circumstances though
(like integration with proprietary webapps or servlets) you might have
to change this parameter.
-->
  <servlet-mapping>
    <servlet-name>Cocoon2</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
</web-app>

```

Finally, to use the XML:DB taglib included with eXist, you should define it as a built-in logicsheet in the configuration of Cocoon. Add the following to WEB-INF/cocoon.xconf:

Example: Adding a new built-in logicsheet to cocoon.xconf

```

<target-language name=" java">
<!-- Defines the XSP Core logicsheet for the Java language -->
  <parameter
    name="core-logicsheet"
    value="resource://org/apache/cocoon/components/language/markup/xsp/java/xsp.xml"
  />
<!-- Insert the following section: -->
  <builtin-logicsheet>
    <parameter name="prefix" value="xmldb"/>
    <parameter name="uri" value="http://exist-db.org/xmldb/1.0"/>
    <parameter name="href" value="resource://org/exist/xmldb.xml"/>
  </builtin-logicsheet>
<!-- ... more builtin-logicsheets ... -->
</target-language>

```

If you want to access eXist through Cocoon's XML:DB pseudo protocol, don't forget to add eXist's XML:DB driver to the <source-handler> section. The section should look something like this:

Example: Adding eXist's XML:DB driver

```
<source-handler logger="core.source-handler">
<!-- ...other predefined protocol handlers... -->
<!-- xmldb pseudo protocol -->
  <protocol name="xmldb" class="org.apache.cocoon.components.source.XMLDBSourceFactory">
<!-- Xindice driver -->
  <driver class="org.apache.xindice.client.xmldb.DatabaseImpl" type="xindice"/>
<!-- Add here other XML:DB compliant databases drivers -->
  <driver class="org.exist.xmldb.DatabaseImpl" type="exist"/>
</protocol>
</source-handler>
```

You should now be able to restart your servlet-engine. You may check if eXist is running by browsing to <http://your-host:8080/your-app/admin>

It is better to remove or comment-out all not needed driver-entries. Some drivers can throw an exception, if they are missing some configuration files or libraries. This causes the Cocoon XMLDBSourceFactory to skip configuring the drivers following the failing one.