Running Apache Tomcat and Apache HTTPD together with AJP13 under FreeBSD

Paul Hoadley, Logic Squad

$Date: 2003/03/31 14:03:55 $

This document describes how to allow the **Apache HTTP Server** to communicate with **Apache Tomcat 4.1.12** via the AJP13 protocol using **mod_jk**. In the author's case, this was used to allow requests to proceed seamlessly from **Apache** to the **Cocoon 2** system running under the **Tomcat** servlet container, but any application running under **Tomcat** can be accessed by this approach. This document assumes that **Tomcat 4.1.12** has been installed. The procedure for doing this is straightforward, and is described elsewhere. This document will assume that **Cocoon 2** has also been installed, but only during the final testing phase. It is actually *not* at all essential that **Cocoon 2** be installed — it is used only as an example. **Apache 1.3.26** was used in the testing of these instructions, though this will certainly work for other versions in the 1.3 line. There seems to be large inter-version differences between **Tomcat** releases, so these instructions may *not* work for versions of **Tomcat** other than 4.1.12. The version of **mod_jk** used was the most current version in the **FreeBSD** Ports System at the time of writing: indicated to be 3.3.1 in the port's `Makefile`, but reported as 1.1.0 by **Apache**'s `server-status` URL. The reason for this difference is not clear.

1. Install Tomcat 4.1.12

This is straightforward using the Ports System and is described elsewhere.

2. Install mod_jk

Again, this requires only building the port. As `root`, execute the following:

```
# cd /usr/ports/www/mod_jk/
# make
# make install
```

At the install phase, this port has an unfortunate dependency on **Tomcat 3.3.1**. Let this port build — it can be de-installed later.

3. Edit configuration files

3.1. Apache configuration files

Edit `httpd.conf` to uncomment the LoadModule and AddModule directives that have been added by the install of **mod_jk**:

```
LoadModule jk_module          libexec/apache/mod_jk.so
```

and:

```
AddModule mod_jk.c
```

Be sure also to remove any references to the older **mod_jserv** module from `httpd.conf`. Add an Include directive to `httpd.conf` that refers to the configuration file we will create below:

```
Include /usr/local/etc/apache/jserv/tomcat-apache.conf
```

Create the file `tomcat-apache.conf` in the directory `/usr/local/etc/apache`:

```
JkWorkersFile /usr/local/jakarta-tomcat4.1/conf/workers.properties
JkLogFile /var/log/mod_jk.log
JkLogLevel info
JkMount /*.jsp ajp13
JkMount /servlet/* ajp13
JkMount /cocoon ajp13
JkMount /cocoon/* ajp13
```

Create the log file for **mod_jk**:

```
# touch /var/log/mod_jk.log
# chown www:www /var/log/mod_jk.log
```

3.2. Tomcat configuration files

Create the file `workers.properties` in the `/usr/local/jakarta-tomcat4.1/conf` directory:

```
workers.tomcat_home=/usr/local/jakarta-tomcat4.1
workers.java_home=/usr/local/jdk1.3.1
ps=/
worker.list=ajp12, ajp13
```

```
worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
```

Change the ownership of `workers.properties`:

```
# chown www:www workers.properties
```

Uncomment this section of `/usr/local/jakarta-tomcat4.1/conf/server.xml`:

```
<!-- Define an AJP 1.3 Connector -->
<Connector className="org.apache.ajp.tomcat4.Ajp13Connector"
port="8009" minProcessors="5" maxProcessors="75"
acceptCount="10" debug="0"/>
```

4. Testing the installation

Stop both **Apache** and **Tomcat**:

```
# apachectl stop
# tomcat41ctl stop
```

Start **Tomcat** and then start **Apache**:

```
# tomcat41ctl start
# apachectl start
```

Wait for a minute or so — **Tomcat** takes a while to begin listening on all the appropriate ports. The test the installation:

```
# lynx http://localhost/cocoon
```

The front page of the **Cocoon 2** installation should appear.