

CocoonAndApache

If you want to connect Cocoon with Apache httpd in order to serve static/legacy content directly from Apache you have four possibilities ATM (apart from the obsolete Jserv):

- mod_webapp
 - mod_jk
 - mod_jk2
 - [ApacheModProxy](#)
-

mod_webapp

mod_webapp communicates via the new WARP protocol, using Tomcats WarpConnector. You can easily mount whole webapps/contexts.

Advantages

- easy setup

Disadvantages

- not as stable as JK1.2
- poor differentiation concerning mounts
- no load balancing
- not suitable for Windows
- not suitable for Jetty

Sample 1

mounting cocoon as <http://host.domain.tld/cocoon/>

```
WebAppConnection warpConnection warp localhost:8008
WebAppDeploy cocoon warpConnection /cocoon/
```

Sample 2

mounting cocoon as <http://host.domain.tld/>

```
WebAppConnection warpConnection warp localhost:8008
WebAppDeploy cocoon warpConnection /
```

- Problem: now EVERYTHING is served by cocoon. No way to serve static/legacy content by Apache!

Docs

- <http://jakarta.apache.org/tomcat/tomcat-4.1-doc/config/connectors.html>
 - <http://jakarta.apache.org/tomcat/tomcat-4.1-doc/config/webapp.html>
-

mod_jk

mod_jk communicates via ajp1.3 protocol using an Ajp13Connector

Advantages

- very stable
- supports load balancing
- works with Jetty

Disadvantages

- long winded setup
- poor syntax in mount directives

- Maintenance of mounts needs changes to httpd.conf

mod_jk is configured by a configuration file named workers.properties:

Sample for workers.properties

```
workers.apache_log=/usr/local/apache2/logs/
workers.tomcat_home=/usr/jakarta/catalina
workers.java_home=/usr/java/current
ps=/
# Define 3 workers, 2 real ajp13 and one being a loadbalancing worker
worker.list=worker1 worker2
# Set properties for worker1 (ajp13)
worker.worker1.type=ajp13
worker.worker1.host=host1.domain.tld
worker.worker1.port=8009
worker.worker1.lbfactor=50
worker.worker1.cachesize=10
worker.worker1.cache_timeout=600
worker.worker1.socket_keepalive=1
worker.worker1.socket_timeout=300
# Set properties for worker2 (ajp13)
worker.worker2.type=ajp13
worker.worker2.host=host2.domain.tld
worker.worker2.port=8009
worker.worker2.lbfactor=50
worker.worker2.cachesize=10
worker.worker2.cache_timeout=600
worker.worker2.socket_keepalive=1
worker.worker2.socket_timeout=300
# Set properties for worker3 (lb) which use worker1 and worker2
worker.worker3.balanced_workers=worker1,worker2
```

in your httpd.conf you put your mount directives like

```
# mounting only the index
JkMount / worker2
# further, for example for mounting all html files
JkMount /*.html worker2
# further, for example for mounting everything in dir
JkMount /dir/* worker2
```

Things that unfortunately **DON'T** work!

```
JkMount */dir/ worker2
JkMount **/dir/ worker2
JkMount /dir1/**/dir/ worker2
JkMount /dir1/*/dir/*/*.html worker2
JkMount /(images|movies|audio)* worker2
```

Docs

- <http://www.onjava.com/pub/a/onjava/2002/11/20/tomcat.html>
- <http://jakarta.apache.org/tomcat/tomcat-4.1-doc/jk2/index.html>

mod_jk2

The very new JK2 communicates via ajp1.3/ajp1.4 using CoyoteConnector+JkCoyoteHandler.

Advantages

- supports load balancing
- supports in-process

- works with Jetty
- supports fast unix-sockets
- fine grained configuration
- easy to maintain mounts (no need to edit httpd.conf)

Disadvantages

- jk2 is not as stable as mod_jk (correct me if I am wrong, I did not try the latest versions)
- lack of documentation
- hard to build from cvs (IMO) - no binaries available AFAIK

mod_jk2 is configured by a configuration file named workers2.properties. There you define hosts, ports, workers, uri mapping etc:

Sample

workers2.properties mounting cocoon as host.domain.tld/

```
[shm]
file=/usr/jakarta/catalina/work/jk2.shm
size=1048576

# Example socket channel, override port and host.
[channel.socket:host.domain.tld:8009]
port=8009
host=127.0.0.1

# define the worker
[ajp13:ministrant.leonid:8009]
channel=channel.socket: host.domain.tld:8009

# Uri mapping
[uri:ministrant.leonid/*]
worker=ajp13: host.domain.tld:8009
context=/cocoon
```

Docs

- http://www.pubbitch.org/jboss/mod_jk2.html
- <http://www.mortbay.com/jetty/doc/modjk.html>
- <http://jakarta.apache.org/tomcat/tomcat-4.1-doc/config/jk2.html>
- <http://jakarta.apache.org/tomcat/tomcat-4.1-doc/jk2/index.html>

Taken from <http://marc.theaimsgroup.com/?l=xml-cocoon-users&m=103910018419021&w=2>

[LeoLeonid](#)

Tomcat Connectors

There are some binaries available, as well as source packages for the various Tomcat connectors. The connectors are now in their own project, called jakarta-tomcat-connectors. Unfortunately there is no home page within jakarta.apache.org that I can find, and the online documentation does not point you to the connector download area.

All of the connectors can be found under the following general location:

- <http://jakarta.apache.org/builds/jakarta-tomcat-connectors>

Explore under this directory for the latest binaries and source packages for all of the connectors.

its_toasted@yahoo.com