

# 1 APACHE HTTPD

## 1.1 INSTALLATION

For Apache Web Server installation instructions, see <http://httpd.apache.org>. Red Hat usually comes with Apache pre-installed. To see if you have Apache installed, try running the following command to find the version of Apache HTTPD:

```
/usr/sbin/httpd -v
```

Once Apache is installed, follow the instructions under **Apache HTTPD: Configuration with Tomcat** in order to configure Apache to communicate with Tomcat..

## 1.2 CONFIGURATION WITH TOMCAT

*These instructions are necessary for the Production server only.* It is recommended to use a combined Apache/Tomcat in Production because it is more secure and performs better when using SSL. Some more reasons why the Apache connection is recommended are laid out at <http://www.linuxgazette.com/issue95/millson.html> under the section "Selecting a Connector". These instructions use the `mod_jk` connector, which is the currently recommended Tomcat connector.

1. First, check to see if `mod_jk` connector is pre-installed (most likely it would be in `/etc/httpd/modules` if it is installed). As `root`:

```
locate mod_jk
```

If there is no response, then it is not installed.

2. If the `mod_jk` connector is not installed, you will need to compile it from source. Follow the directions laid out under **Apache HTTPD: Install mod\_jk from Source**. *Note: if you upgrade Tomcat and the connector no longer works, you may need to recompile the `mod_jk` connector from source!*
3. Once the `mod_jk` connector has been installed, you will have to configure Apache to use this connector to communicate with Tomcat. First, modify the existing `workers.properties` file (should be in `/etc/httpd/conf`):

You will need to modify the following Tomcat and Java home directories:

```
workers.tomcat_home=[ tomcat ]
```

```
workers.java_home=[ java ]
```

In addition, you may need to uncomment the JVM for Unix:

```
# Unix - Sun VM or blackdown
```

```
worker.inprocess.jvm_lib=$(workers.java_home)$ (ps) jre$ (ps)
)lib$ (ps) i386$ (ps) server$ (ps) libjvm.so
```

*Note: initially the path above was*

```
[java]/jre/lib/i386/classic/libjvm.so
```

*However, the correct path of the libjvm.so (at least for RedHat) is*

```
[java]/jre/lib/i386/server/libjvm.so.
```

4. Next, you need to create a configuration file for the `mod_jk` module. In the `/etc/httpd/conf.d/` directory, create a file called `jk.conf` which has the following content:

```
#
# Use the JK Module to connect to Tomcat Instance
# (currently only used for the root / context)
#

# Load mod_jk module
LoadModule      jk_module  modules/mod_jk.so

# Where to find workers.properties
JkWorkersFile   /etc/httpd/conf/workers.properties

# Where to put jk logs
JkLogFile       /var/log/httpd/mod_jk.log

# Set the jk log level [debug/error/info]
JkLogLevel      info

# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "

# JkOptions indicate to send SSL KEY SIZE,
JkOptions        +ForwardKeySize +ForwardURISCompat -
                ForwardDirectories

# JkRequestLogFormat set the request format
JkRequestLogFormat "%w %V %T"

# Send all requests for root context / to worker ajp13
# Note: ajp13 is defined in workers.properties and
# uses the AJP 1.3 Protocol
JkMount  /* ajp13

#Deny direct access to any WEB-INF and META-INF directories
<LocationMatch "/WEB-INF/">
AllowOverride None
Deny from all
</LocationMatch>
```

```
<LocationMatch "/META-INF/">
AllowOverride None
Deny from all
</LocationMatch>
```

5. Next, you need to take a look at the Tomcat server.xml configuration file. Ensure that the following Connector is uncommented:

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" URIEncoding="UTF-8"
    tomcatAuthentication="false" enableLookups="false"
    redirectPort="8080" protocol="AJP/1.3" />
```

Make sure that the `redirectPort` specified corresponds to the port you defined for the `ajp13` worker (this port number is defined in the `workers.properties` file). This port should be either 8443 (for Tomcat with SSL) or 8080 (for normal Tomcat). In addition, make sure the `URIEncoding` is set to UTF-8, and `tomcatAuthentication` is set to false (since authentication will be done in Apache instead).

6. Restart Tomcat and Apache (as `root`):

```
service httpd stop
~dspace/bin/stoptomcat
~dspace/bin/starttomcat
service httpd start
```

7. Test the connection between Apache and Tomcat. You should now be able to get to DSpace whether you specify port 8080 (for Tomcat) or not. For example, the following URLs should bring you to the same DSpace:

<http://my-host-name:8080/dspace>  
<http://my-host-name/dspace>

In addition, if you have Apache running with SSL, you should be able to access DSpace across SSL (although currently, the default certificate will be signed for `localhost.localdomain`):

<https://my-host-name/dspace>

### 1.3 INSTALL MOD\_JK FROM SOURCE

1. Login as the `root` user.
2. Before trying to build `mod_jk`, you must make sure you have the following programs installed (use the `which` command to check for each):

```
libtool (ftp://ftp.gnu.org/gnu/libtool)
autoconf (http://ftp.gnu.org/gnu/autoconf)
ant (http://jakarta.apache.org/ant/)
```

ant should already be installed. If libtool or autoconf are missing (both should be in /usr/bin), download the source and compile using the following commands:

```
./configure
make
make install
```

3. In addition, you must have the Apache Web Server development tools installed. A quick way to check for this is to check for the APache eXtenSion tool (apxs)

which apxs (should be in /usr/sbin, if installed)

If apxs is missing, you can use the following command in RedHat to install the httpd-devel RPM (as root):

```
up2date -i httpd-devel
```

4. Download the latest mod\_jk source from the Tomcat Download site [http://jakarta.apache.org/site/downloads/downloads\\_tomcat.html](http://jakarta.apache.org/site/downloads/downloads_tomcat.html).

5. Unzip the contents into your home directory:

```
gunzip -c jakarta-tomcat-connectors-1.2.14.1-src.tar.gz |
tar -xvf -
```

6. Configure the connectors with the path to the apxs file on your system:

```
cd jakarta-tomcat-connectors-1.2.14.1-src
cd jk/native
./configure --with-apxs=/usr/sbin/apxs
```

7. Build mod\_jk with the following command:

```
make
```

8. Assuming all went well, the mod\_jk.so file will be created in the apache-2.0 subdirectory. You need to copy this file to Apache's shared object files directory (e.g. /etc/httpd/modules). From the same jk/native directory run the following:

```
cp apache-2.0/mod_jk.so /etc/httpd/modules
```

9. In addition, copy the sample workers.properties file to the Apache configuration directory. Assuming you are still in the jk/native directory, run the following commands:

```
cd ../conf
cp workers.properties /etc/httpd/conf
```