Building a Secure Web Server

Jason Novotny and Marcia Perry

Distributed Systems Department

Ernest Orlando Lawrence Berkeley National Laboratory

June 4, 2001

Organization

- Major Components
- Overview of the Build Process
- Configuration
- URL to File System Mappings
- Starting/Stopping Web Server Processes
- More Information

Web Server Components

- **Web server**: application that responds to HTTP requests by returning 'web' resources (e.g., HTML files, images, applets,CGI output, …) over the Internet
- **Servlet container** (or **servlet engine**): runtime shell that invokes servlets on behalf of clients; software that runs servlets and manages them through their lifecycle

Servlet Containers

Servlet containers can be partitioned as:

- Standalone: Integral part of web server (as when using a Java-based web server)
- Add-on component to web server: Java container implementation + web server plugin
  - Servlet code runs inside Java container
  - Java container runs inside of JVM
  - Web server plugin opens JVM

Apache and Tomcat

Apache: "Industrial strength" HTTP/1.1 compliant web server

- Highly configurable
- Implements many features in addition to the core functionality (e.g., security/access control, virtual hosting, CGI script execution, …)
- Extensible with third-party modules (e.g., servlet engine, security, WebDAV, …)

Apache and Tomcat

Tomcat: Java-based servlet container w/ JSP environment

- Execution modes:
  - **Standalone**: default mode for Tomcat
  - **In-process add-on**: web server plugin opens JVM inside web server's address space; plugin passes servlet/JSP requests to servlet container via JNI
  - **Out-of-process add-on**: web server plugin opens JVM outside web server; plugin and JVM communicate using IPC mechanism (TCP/IP sockets and)

Tomcat Execution Modes

Standalone

- Not as fast as Apache for static pages
- Not as configurable as Apache
- Not as robust as Apache
- May not support functionality found only in Apache modules (e.g., Perl, PHP, security)
- Mainly for development and debugging

Tomcat Execution Modes

In-process add-on

- Suitable for multi-threaded single-process servers
- Provides good performance
- Limited in scalability

Tomcat Execution Modes

Out-of-process add-on

- Poorer response time than for in-process servlet container
- Better scalability
- Better stability

Tomcat and Apache

Communication mechanism between Tomcat and Apache:

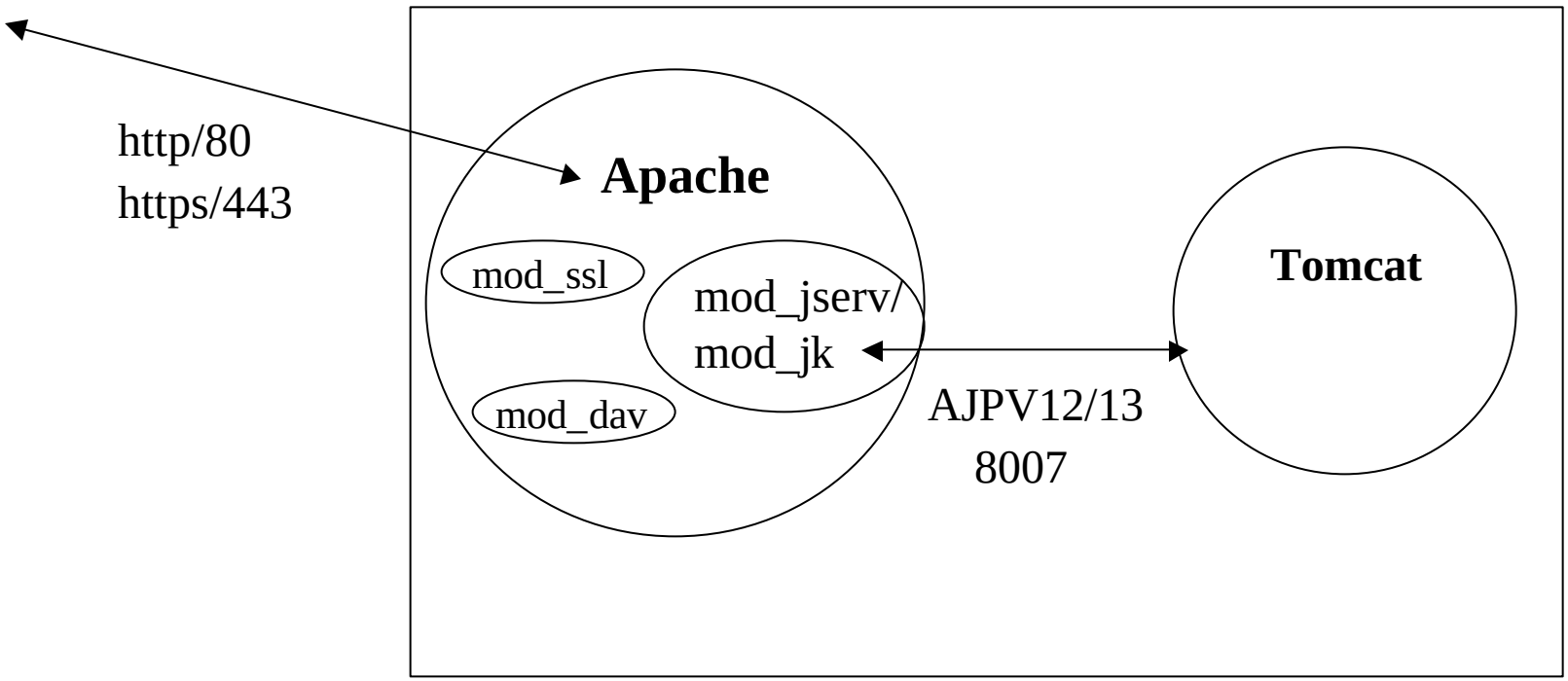
- Termed “web server adapter” or “connector”
- Implemented as library (e.g., mod_jserv.so, mod_jk.so)
- Uses/manages TCP connections
- Uses the AJPV12/AJPV13 communication protocol

Tomcat vs. Jserv

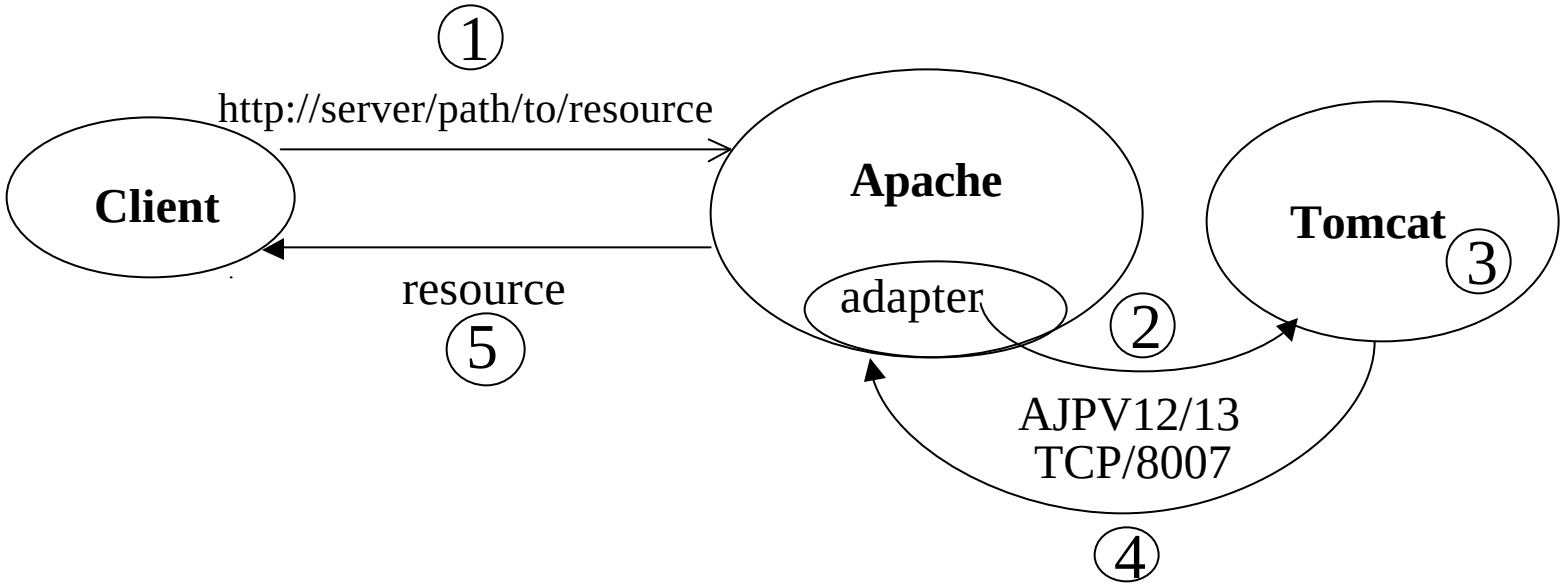
Tomcat's mod_jserv != Apache Jserv

- Jserv for Apache (www.apache.org/jserv)
  - Older; in maintenance-only mode
  - Servlet API 2.0-compliant container
- Tomcat's mod_jserv
  - Servlet API 2.2 and JSP 1.1-compliant container
  - Supports Apache, IIS, and Netscape servers

Our Basic Installation



How Apache & Tomcat Interoperate



Apache in standalone mode; Tomcat in out-of-process add-on mode

Building and Installing Apache and Tomcat

- Apache supports statically-linked and dynamically-linked modules (DSOs)
- Our builds were done under Solaris 2.7 and Linux Redhat 6.2
- Script to automate the build/configure process available at www-itg-lbl.gov/Grid/projects/WebServer-SG.html
- Step-by-step procedure available at www-itg.lbl.gov/Private/apache_build.html

Building and Installing Apache and Tomcat

- Our components:
  - Binary distribution of Tomcat
  - Apache built from source
  - Statically-linked Apache modules (mod_access, mod_cgi, mod_so, mod_dav, …)
  - Dynamically-linked Apache modules (mod_ssl, mod_jserv, …)

Building and Installing Apache and Tomcat

Assumptions:

- Java already installed (JDK 1.2/JDK1.3)
- APACHE = /usr/local/apache
- TOMCAT = /usr/local/tomcat

1. Build OpenSSL (needed for mod_ssl)
2. Build optional MM shared memory library
3. Configure mod_ssl (build in step 6)
4. Build mod_dav

Building and Installing Apache and Tomcat

5. Build and install Apache w/ DSO support, mod_ssl, and mod_dav.

Gotcha: Docs describe 2 ways to configure—in APACHE/src/ w/ ‘Configure’ (APACI method) or in APACHE/ w/ ‘configure’. The latter worked better!! See the INSTALL file in the top-level APACHE directory of the source distribution.

Building and Installing Apache and Tomcat

5. Build and install Apache (cont'd.)

Gotcha: In addition to its binary (httpd), apache builds tools, one of which is 'apxs', used to build shared objs. If Apache isn't built w/ DSO support, you will get an error like this when building *.so:

apxs: Break: Command failed with rc=16711680

Solution: Include the following args to 'configure':

--enable-module=so –enable-rule=SHARED_CORE

Building and Installing Apache and Tomcat

5. Build and install Apache (cont'd.)

Gotcha: In building mod_ssl, you'll need to make a certificate. You can make a temporary certificate for a quick build and testing, but remember to get a real certificate later! See https://idcg-ca.lbl.gov and click 'SSL Server' link. Put certs in APACHE/conf/ssl.* dirs.

Building and Installing Apache and Tomcat

6. Build the Tomcat's mod_jserv.so connector module for Apache

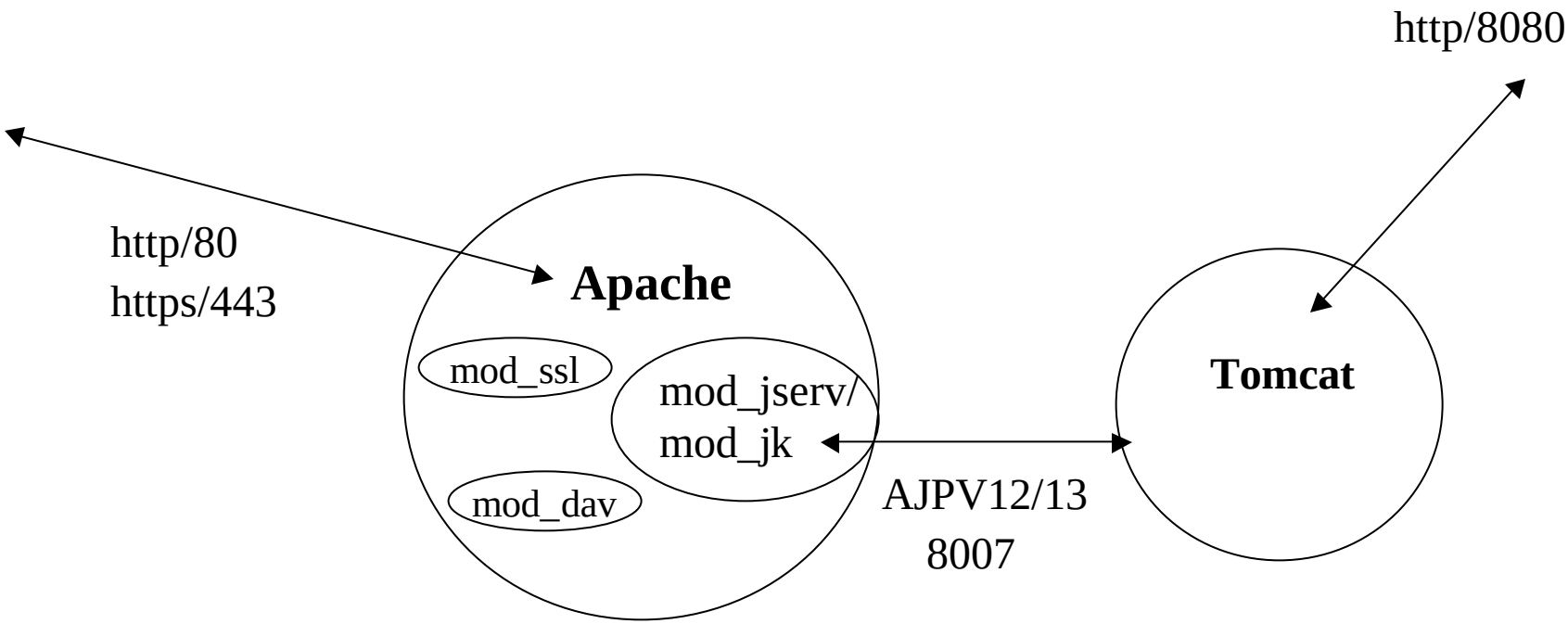
Gotcha: Since the build is done in the Tomcat src tree, be sure to copy autochange.so and mod_jserv.so into Apache's libexec/ directory!

Configuration

- **Apache**:
  - **httpd.conf** (in APACHE/conf/): master config file
  - **tomcat-apache.conf** (generated by tomcat): included in httpd.conf for mod_jserv
- **Tomcat** (in TOMCAT/conf/):
  - **server.xml**: global config file
  - **tomcat.conf**: lets web server work with Tomcat
  - **web.xml**: configures Tomcat contexts

Configuring the Ports

Default configuration



Configuring the Ports

server.xml

```
<!– disable webserver on port 8080
    <Connector className="org.apache.tomcat.service.SimpleTcpConnector">
        <Parameter name="handler"
value="org.apache.tomcat.service.http.HttpConnectionHandler"/>
        <Parameter name="port" value="8080"/>
    </Connector>
-->
    <Connector className="org.apache.tomcat.service.SimpleTcpConnector">
        <Parameter name="handler"
value="org.apache.tomcat.service.connector.Ajp12ConnectionHandler"/>
        <Parameter name="port" value="8007"/>
    </Connector>
```

Configuring the Ports

tomcat.conf

```
#Tell Apache to load the shared object communication module
LoadModule jserv_module libexec/mod_jserv.so

# Set communication protocol and port
ApJServDefaultProtocol ajpv12
ApJServDefaultPort 8007
```

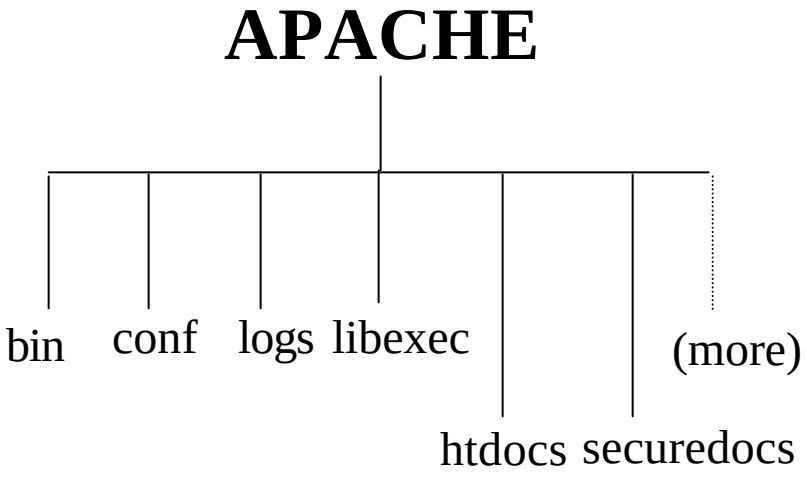
Configuring the Ports

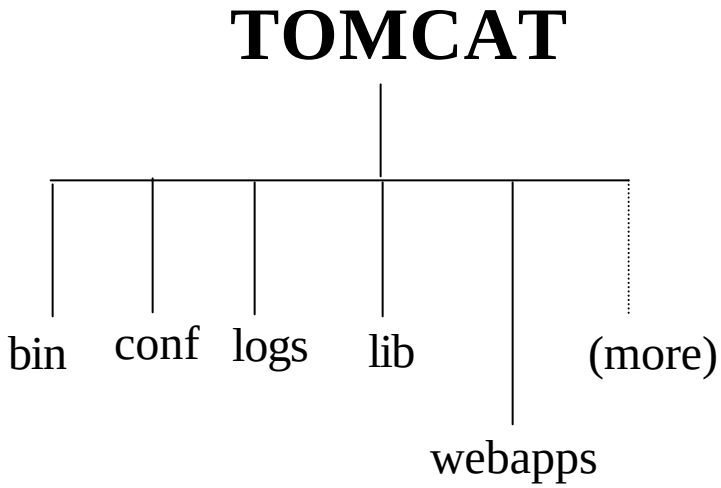
httpd.conf

```
ServerRoot “/usr/local/apache”
# Here’s where we can overwrite default ports
Port 80
<IfDefine SSL>
Listen 80
Listen 443
</IfDefine>

<VirtualHost _default_:443>
```

Sample File System

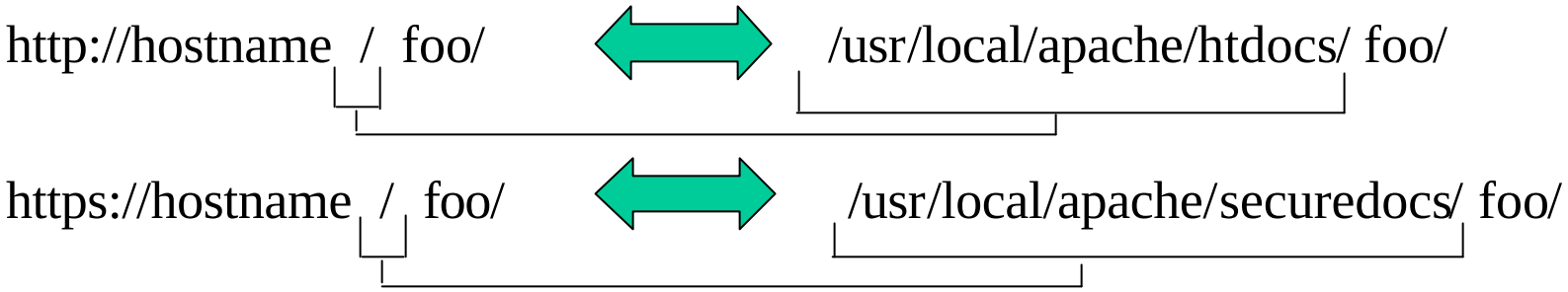




URL to File System Mappings

httpd.conf

```
DocumentRoot "/usr/local/apache/htdocs"
<IfDefine SSL>
# General setup for the virtual host
DocumentRoot "/usr/local/apache/securedocs"
# Lots of stuff
</IfDefine>
Include /usr/local/tomcat/conf/tomcat-apache.conf
```



Apache Directory Access

Restrict access on per-directory basis via **httpd.conf**.

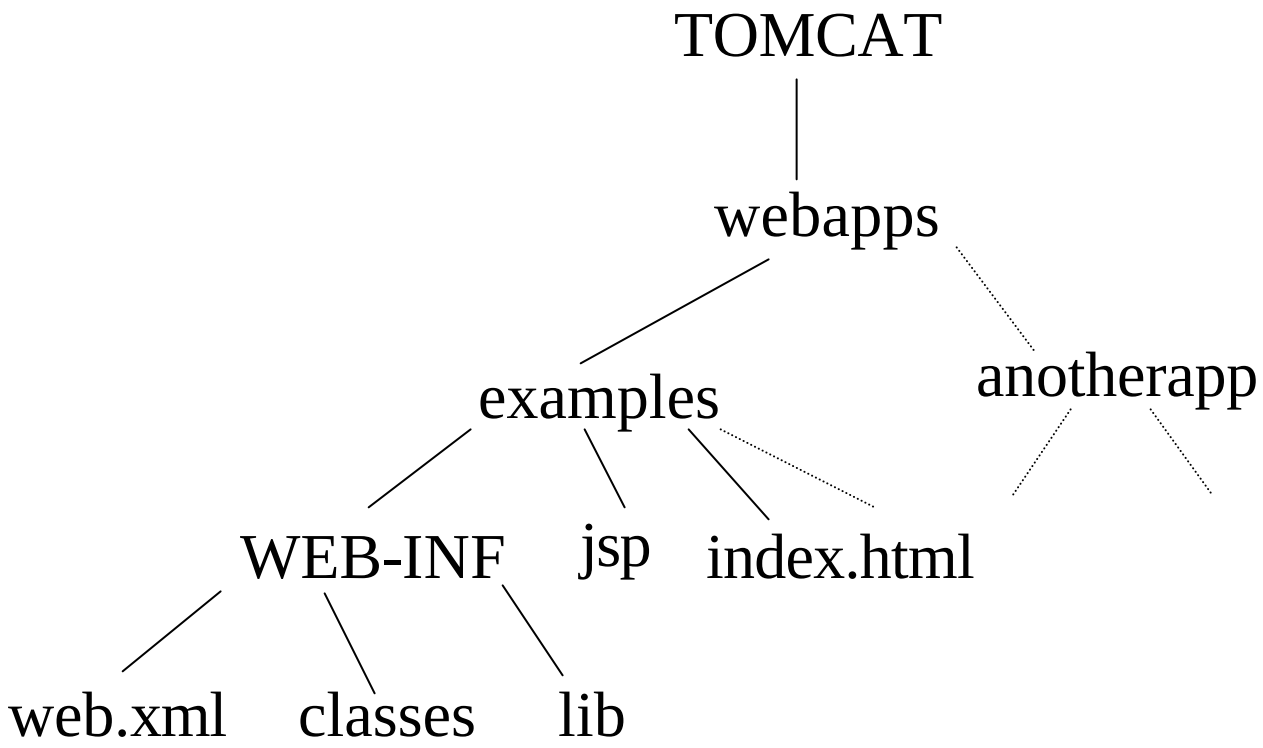
```
<Directory />
    AllowOverride None
</Directory>
<Directory “/usr/local/apache/htdocs/webDAVdir”>
    Order deny, allow
    Deny from all
    Allow from .lbl.gov
    DAV On
</Directory>
```

Apache Directory Access

Per-directory access restriction (httpd.conf)

```
<Directory "/usr/local/apache/htdocs/webDAVdir">
    Order deny, allow
    <Limit GET POST >
        Deny from all
        Allow from .lbl.gov
    </Limit>
    <Limit PUT DELETE MKCOL COPY MOVE LOCK UNLOCK>
        Deny from all
        Allow from 131.243.2
    </Limit>
</Directory>
```

Tomcat File System



URL to File System Mappings

tomcat-apache.conf

```
AddType text/jsp .jsp
AddHandler jserv-servlet .jsp
Alias /examples /usr/local/tomcat/webapps/examples
ApJServMount  /examples/servlet  /examples
<Location /examples/WEB-INF/ >
    AllowOverride none
    deny from all
</Location>
ApJServMount /servlet /ROOT
```

URL to File System Mappings

server.xml:

```
<Context path="/examples"
docBase="webapps/examples" debug="0"
reloadable="false"
</Context>
```

SIDE NOTE: Tomcat docs recommend turning on **servlet auto-reloading only for development**. However, specifying reloadable="true" did not seem to work. When a servlet was recompiled, Tomcat had to be restarted.

Configuring a Context

web.xml

```
<web-app>
        <servlet>
                <servlet-name>MyServlet</servlet-name>
                <servlet-class>SimpleServlet</servlet-class>
        </servlet>
        <servlet-mapping>
                <servlet-name>MyServlet</servlet-name>
                <url-pattern>/servlet/*</url-pattern>
        </servlet-mapping>
</web-app>
```

Starting Apache

- Specify user and group to run as (in httpd.conf):

  User nobody

  Group cpc

- Remember to add libexec/ to LD_LIBRARY_PATH
- Start Apache as root:

  # cd /usr/local/apache/bin

  # ./apachectl startssl

Starting Apache

Sample startup script: APACHE/start

```
#!/bin/sh
LD_LIBRARY_PATH=/usr/local/apache/libexec:/usr/local/openssl-0.9.6:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
/usr/local/apache/bin/apachectl startssl
echo “Apache started”
```

Starting Apache

Usage: APACHE/bin/httpd [-d directory] [-v] [-h] [-l]…

-d: specify alternative ServerRoot

-v: show version number

-h: list available command line options

-l: list compiled-in (static) modules

Starting Tomcat

- Do NOT start Tomcat as root.
  - Create a new user account or use an existing one.
- Use the ‘startup.sh’ script in TOMCAT/bin
- If necessary add or modify entries for JAVA_HOME, TOMCAT_HOME, and CLASSPATH.

Tomcat Startup Script

TOMCAT/bin/startup

```
#!/bin/sh
TOMCAT_HOME=/usr/local/tomcat
export TOMCAT_HOME
PATH=/usr/local/java/bin:$PATH
export PATH
CLASSPATH=$CLASSPATH:/usr/local/MyJavaPkg:.
export CLASSPATH
BASEDIR=`dirname $0`
$BASEDIR/tomcat.sh start “$@”
```

Stopping Apache/Tomcat

- Tomcat
  - As ‘tomcat user’ run TOMCAT/bin/shutdown.sh
- Apache
  - As root, use apachectl (or write a ‘stop’ script):

```
# cd /usr/local/apache/bin
# ./apachectl stop
        OR
# cd /usr/local/apache
# ./stop
```

More Information

- Wainright, P., “Professional Apache,” Wrox Press Ltd.
- http://www.webdav.org/
- http://httpd.apache/org/docs/
- http://jakarta.apache.org/tomcat/
- http://java.sun.com/products/servlet/2.2/

  download Java Servlet Specification, v 2.2

More Information

- http://jakarta.apache.org/tomcat/jakarta-tomcat/src/doc/
  - uguide/tomcat_ug.html (Tomcat – A Minimalistic User's Guide)
  - tomcat-apache-howto.html (Tomcat-Apache HOWTO)
  - mod_jk-howto.html (Working with mod_jk)
  - Tomat FAQ (from links in above pages)

More Information

- http://www.ccl.net/cca/software/UNIX/apache/
  tomcat3.1b1-faq.html, README T1SB.shtml, solaris-t3.2/README.shtml
- http://www-itg.lbl.gov/Private/apache_build.html
- http://www-itg.lbl.gov/Grid/projects/WebServer-SG.html
- https://idcg-ca.lbl.gov

My email addr: MPerry@lbl.gov