Tomcat/Apache for Solaris sparc

Installation

1. Install Java2 SDK v1.3.1 from java.sun.com (`j2sdk-1_3_1_02-solsparc.sh` -about 25MB in size)
   as root:
   ```
   cd /usr
   /home/download/j2sdk-1_3_1_02-solsparc.sh
   ```
   make /usr/java a symlink to the new Java SDK :
   ```
   rm java
   ln –s j2se java
   ```

2. Download Tomcat v4.0.3 (binary version, not source) from
   http://jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.0.3/ (about 4.1MB)
   As root, untar it using gnu tar (you may need to download gnu tar from www.sunfreeware.com) into /opt/jakarta-tomcat-4.0.3
   Then change it to be owned by a non-root user :
   ```
   chown –R bob /opt/jakarta-tomcat-4.0.3
   ```

3. As that non-root user, start up Tomcat:
   ```
   export JAVA_HOME; JAVA_HOME=/usr/java
   export CATALINA_HOME; CATALINA_HOME=/opt/jakarta-tomcat-4.0.3
   $CATALINA _HOME/bin/startup.sh
   ```
   You should now be able to connect to port 8080 of your server using your favourite browser, and view the Tomcat examples.

4. Download Apache v1.3.23 (source version) from apache.org : untar it into e.g. /export/home/apache-1.3.23 ; then compile it :
   ```
   ./configure –-enable-rule=SHARED_CORE --enable-module=so
   make
   make install
   ```
   (this will install into /usr/local/apache)
   modify /usr/local/apache/conf/httpd.conf :
   - make sure that ServerName is set to something that is resolvable by your client machines. You can use the server's IP address if all else fails.
   - Add the following lines at the end of httpd.conf:
     ```
     LoadModule webapp_module libexec/mod_webapp.so
     WebAppConnection warpConnection warp localhost:8008
     WebAppDeploy active warpConnection /examples
     ```

5. Before starting up Apache, we need to install the mod-webapp DSO (Tomcat plug-in module for Apache): download a binary version for Solaris sparc from
   http://jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.0/bin/webapp-module-1.0-tc40-sol8sparc.tar.gz
   untar it using gnu tar (you can get a copy of gnu tar from sunfreeware.com) and copy the file `mod_webapp.so` into `/usr/local/apache/libexec`.

   Now check the Apache config file is OK:
   ```
   /usr/local/apache/bin/apachectl configtest
   ```
   It should report "`Syntax OK`" : ignore any warning about mod-webapp already being loaded.

6. Now start up Apache (`/usr/local/apache/bin/apachectl start`) and you should be able to see the Tomcat examples via `http://yourservername/examples/`

Some tips

Tomcat runs as a single Unix process: a Java virtual machine which listens on port 8008 for requests from Apache. Apache passes all requests it receives which start with the string '/examples/' to Tomcat. This keyword (examples) can be modified to suit your application.

The important Tomcat config file is `/opt/tomcat/conf/server.xml`
This contains all the definitions for the 'examples' web-application. It also defines the Tomcat internal web server on port 8080, and an AJP connector on port 8009, both of which you may want to comment out in a production environment.

Tomcat processes JSP scripts via a special servlet which is invoked when it sees an incoming request ending in .jsp. (this is configurable in the `/opt/tomcat/conf/web.xml` configuration file.

To change Tomcat so that it uses a different URL keyword instead of 'examples' (e.g. 'active'):

- stop Tomcat (`$CATALINA_HOME/bin/shutdown.sh`)
- edit `/opt/jakarta-tomcat/conf/server.xml` and change occurrences of the word 'examples' to your chosen keyword (e.g. 'active'),
- copy the contents of `/opt/jakarta-tomcat/webapps/examples` into `/opt/jakarta-tomcat/webapps/active` (you can remove the example scripts and servlets later).
- start Tomcat (`$CATALINA_HOME/bin/startup.sh`)
- modify the Apache httpd.conf so that the Tomcat WebbAppDeploy line reads:
  `WebAppDeploy active warpConnection /active`
- restart Apache

If you ever see an error in your browser "Web-application not deployed yet" when trying to access a Tomcat-based page, it means that Tomcat isn't started up fully yet (this can take several minutes). To see if Tomcat's JVM has started up fully, use:

```
bob $    /usr/ucb/ps auxww|grep tomcat
root 16718 0.1 /usr/java/bin/../bin/sparc/native_threads/java -Djava.endorsed.dirs=/usr/opt/
jakarta-tomcat-4.0.3/bin:/usr/opt/jakarta-tomcat-4.0.3/common/lib
```

The 0.1 shown above shows the % of the CPU that Tomcat is currently using; this will be more than 2% until tomcat is fully started up, so wait until it falls below 2% before starting Apache or trying to acccess JSP scripts via a browser.

Getting the Send Mail JSP example to work:

1. For some reason the SendMail servlet isn't compiled in the binary tomcat distribution :
   To compile it:
   add `/usr/java/bin` to your PATH
   `cd /opt/tomcat/webapps/examples/WEB-INF/classes`
   `export CLASSPATH;`
   `CLASSPATH=/opt/tomcat/common/lib/servlet.jar:/opt/tomcat/common/lib/mail.jar`
   `javac SendMailServlet.java`

2. There is also a typo in the JSP script that calls the sendmail servlet :
   edit `/opt/tomcat/webapps/examples/jsp/mail/sendmail.jsp`
   and about line 22, add `/servlet` as shown :
   `<form method="POST" action="../..`**`/servlet`**`/SendMailServlet">`