Installing Jakarta-Tomcat with Apache on Debian

Matthew Danish

5/25/2001

Contents

1 Introduction

Installing the Jakarta Tomcat servlet container can be quite a pain, as I discovered, even on a convenient distribution such as Debian. I was able to make Tomcat from unstable work on a testing system using the pinning feature, however. The Debian package took care of most of the dirty work, but the configuration files were not quite tailored correctly for my system.

In this document I am assuming that you have a working Apache installation (I used 1.3.19 from testing) and that you have APT 0.5.3 and dpkg 1.9.6 or greater. I am running Kernel 2.2.19, and the installation was conducted on 'woody' but using certain packages from 'unstable' (such as Tomcat itself).

This document was written when potato (2.2) was 'stable' and woody was 'testing'. If things have changed by the time you read this, packages may have been updated, but the general gist should be of use.

In addition, please consult the documentation at `http://jakarta.apache.org/` for an authoritative reference. In no way do I guarentee that the instructions in this document are entirely accurate, they are just a reflection of my experience.

1.1 What is Tomcat and how does it interact with Apache?

Tomcat is, in this configuration, a stand-alone TCP/IP server that communicates with Apache via mod_jk. It can also act as a simple HTTP server in it's own right, and there are ways of getting it to run within the Apache process as well. This document does not cover that however. For a better description of the whole Apache->mod_jk->Tomcat process, see `http://www.ccl.net/cca/software/UNIX/apache/tomcatfaq.shtml`.

2 The Packages

Obtaining tomcat:

apt-get install tomcat/unstable[1] libapache-mod-jk

This should install Tomcat and the Apache module mod_jk, as well as jikes (a Java compiler), libxerces-java, and libservlet2.2-java. It will probably install jdk1.1 by default, but you have a choice of j2sdk1.3, j2re1.3, ibm-jdk1.1-installer, and ibm-jre1.1-installer (as of this date). For more information on using j2sdk1.3 with Tomcat, please see Section 6.

[1] If, by the time you read this, tomcat is in testing or stable, you can probably omit "/unstable"

2.1 A quick test

Point your web browser at `http://localhost:8081/tomcat-examples/`. Tomcat is also a simple web server as well as a servlet container, so you should see a web page pop up. Click on servlets/ and try out some of them. If it takes a very long time for Tomcat to respond, then read on.

3 Configuration

3.1 /etc/apache/httpd.conf

The installation scripts for libapache-mod-jk will insert the necessary lines into here, but I don't like the way they do it, so here's what I do:

I add "LoadModule jk_module /usr/lib/apache/1.3/mod_jk.so" somewhere before "LoadModule rewrite_module /usr/lib/apache/1.3/mod_rewrite.so"

The scripts will place a section that looks like this in the file:

```
<IfModule mod_jk.c>
# The following line is for apacheconfig - DO NOT REMOVE!
JkWorkersFile /etc/apache/mod_jk/workers.properties
Include /etc/apache/mod_jk/tomcat-auto
</IfModule>
```

If it's located near the end of the file, it's fine. Otherwise that might be a good location for it. Note the "Include" line, that file is an automatically generated file that is outputted by Tomcat.

3.2 /etc/apache/mod_jk/tomcat-auto

This is an automatically generated file, generated when you start Tomcat[2]. You will probably want to copy it to another file, such as "/etc/apache/mod_jk/tomcat.conf" and remember to change the "Include" line in httpd.conf to reflect this.

When you do copy it, you will probably want to make the following changes inside it:

- Comment out "LoadModule jk_module libexec/mod_jk.so"
  - You already have it in httpd.conf, and that path is wrong anyway.

[2] It may take a second or two to appear, while Java starts up

- Take a look at the auto configuration sections near the bottom of the file. There should be two by default "/tomcat-examples" and "/examples". Use these as the models for adding new contexts, thought you might want to comment them out.

Here is a sample context from "tomcat-auto":

```
Alias /tomcat-examples "/usr/share/tomcat/webapps/tomcat-examples"
<Directory "/usr/share/tomcat/webapps/tomcat-examples">
  Options Indexes FollowSymLinks
</Directory>
JkMount /tomcat-examples/servlet/* ajp13
JkMount /tomcat-examples/*.jsp ajp13
# The following line prohibits users from directly accessing WEB-
INF
<Location "/tomcat-examples/WEB-INF/">
  AllowOverride None
  deny from all
</Location>
# The following line prohibits users from directly accessing META-
INF
<Location "/tomcat-examples/META-INF/">
  AllowOverride None
  deny from all
</Location>
```

As you can see, these are all Apache directives. If you are unsure of their meaning, please consult the documentation at `http://www.apache.org/`. JkMount is the mod_jk.so specific directive that identifies to mod_jk what requests should be redirected to Tomcat. ajp13 is the protocol that is used to make this request. ajp12 is an older protocol, that will be used in the examples, in the above I changed it to ajp13 and you should probably too.

3.3 /etc/tomcat/server.xml

This is the main configuration file for Tomcat. Most of it should be set correctly and can be left alone. Please note the section:

```
<Connector className="org.apache.tomcat.service.PoolTcpConnector">
  <Parameter name="handler"
  value="org.apache.tomcat.service.http.HttpConnectionHandler"/>
  <Parameter name="port"
  value="8081"/>
</Connector>
```

This configures Tomcat to be an HTTP server on port 8081. You probably do not want this in a production environment, it is a security risk, so comment it out once you have set everything up. Comments are done like in HTML, with <!– ... –>.

Near the bottom of the file is the Context section. Here is where you add new web applications.

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

The path defines the URL path that this context is for. In this example, that would be `http://localhost/examples/`. docBase specifies where the web application is to be found. You may specify absolute paths, but if you put a relative path it will be relative to TOMCAT_HOME which is /usr/share/tomcat in Debian, by default. crossContext allows the web app to access other contexts, default is true. Default for debug is 0. reloadable="true" will cause the web application to be recompiled each time it is accessed, which may be bad for a production environment; default is "true" however.

When you add a Context to this file, you must update the tomcat.conf that you copied from tomcat-auto, if you decided to copy it. When you do Virtual Hosts (see Section 7), you must add the information that would normally go in tomcat-auto to the correct <VirtualHost> section instead.

3.4 /etc/apache/mod_jk/workers.properties

A worker is basically a Tomcat listener, it waits for information from Apache and gives it to Tomcat. This file comes with all the "inprocess" workers uncommented, the "inprocess" standing for "in the Apache process" which is not what we want to do. Here is my workers.properties (note that the file comes with extensive comments):

```
workers.tomcat_home=/usr/share/tomcat
workers.java_home=/usr/lib/jdk1.1/
ps=/
worker.list=ajp12,ajp13
worker.ajp12.port=8007
worker.ajp12.host=localhost
worker.ajp12.type=ajp12
worker.ajp12.lbfactor=1
```

worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
worker.ajp13.lbfactor=1
worker.loadbalancer.type=lb
worker.loadbalancer.balanced_workers=ajp12,ajp13
worker.inprocess.class_path=$(workers.tomcat_home)$(ps)classes

And all other definitions are commented out.

4 Checking out your configuration

Run /etc/init.d/tomcat restart, and if you are still using /etc/apache/mod_jk/tomcat-auto then wait a few seconds, before running /etc/init.d/apache reload.

Try to access `http://localhost/tomcat-examples/` and check out the servlets/ directory and the examples there. If it works, great, if not, check your configuration.

5 Security

Tomcat listens on port 8081 (HTTP), 8007 (ajp12), and 8009 (ajp13). You will probably want to firewall off 8007 and 8009 from remote hosts and disable Tomcat's HTTP server (see Section 3.3).

6 Using j2sdk1.3

You can use j2sdk1.3 with Tomcat as well. Add this following line to your /etc/apt/sources.list:

deb `ftp://ftp.uk.linux.org/pub/linux/java/debian/` woody non-free

Or go to `http://www.blackdown.org/` and find a mirror.

Then you can: apt-get install j2sdk1.3

Once you've done this, edit "/etc/apache/mod_jk/workers.properties" and change the workers.java_home to /usr/lib/j2sdk1.3/. Then restart Tomcat following the instructions in Section 4.

7 Virtual Hosts

Tomcat is capable of working with Apache's virtual hosting. Just like when you add a context, you must modify server.xml, but this time you put the Context within <Host> </Host> tags, such as in this example:

```
<Host name="some.domain.com" >
<Context path=""
 docBase="webapps/examples" />
<Context path="/examples"
 docBase="webapps/ROOT" />
</Host>
```

As you can see, other than the <Host> </Host> tags, it's similar to adding an ordinary Context entry.

Now in /etc/apache/httpd.conf you must configure the Apache side of things. Inside of the <VirtualHost> </VirtualHost> section you must place the Apache directives that normally would have gone in /etc/apache/mod_jk/tomcat.conf or tomcat-auto. An example of this is given in Section 3.2.