

## Installing Tomcat 6 on Ubuntu 8.04 (Hardy)

**Miles Jordan**

**AADC**

Last Updated Wednesday, 22 December 2010



## 1. APT Configuration

Tomcat 6 is not provided by the standard Hardy package repositories, but it can be retrieved from the backports repository. In order to allow Ubuntu access to that repository, edit `/etc/apt/sources.list` and uncomment the *backports* repositories.

Next you should run an update on APT to download package information from those repositories:

```
# apt-get update
```

## 2. Installation

Now you should be able to install Tomcat6:

```
# aptitude install tomcat6
```

If you encounter any errors, you should purge any tomcat 6 installations, which will remove any old configuration files that might be lying around, and also remove any references to tomcat6 from the `/etc/init.d` directory and `/etc/rc.*` directories.

```
# aptitude purge tomcat6 tomcat6-admin tomcat6-examples
# rm /etc/**tomcat*
```

### 2.1. Tomcat Manager

You should also install the tomcat manager, which allows you to view and configure webapps:

```
# aptitude install tomcat6-admin
```

### 2.1. Java Fonts

Geoserver will only have one font available to it unless extra fonts are installed into the JRE:

```
# aptitude install msttcorefonts ttmkfdir
# cd /usr/lib/jvm/jdk/jre/lib/fonts
# cp /usr/share/fonts/truetype/msttcorefonts/* ./
# ttmkfdir -o fonts.dir
```

## 3. Tomcat Configuration

### 3.1. Security Manager

You should disable the tomcat security manager, otherwise some webapps will not be able to work properly and logging will not work. NOTE: We do this because we don't know how to configure it properly. Someone should figure that out!



Search for SECURITY in `/etc/init.d/tomcat6` and change it to:

```
TOMCAT_SECURITY=no
```

### 3.2. Users

You also need to set up the admin user that can access the manager webapp. You need to choose a password for the aadc user. In this example we use “yourpass”, but you should use a more appropriate aadc password.

Edit `/etc/tomcat6/tomcat6-users.xml` to look like:

```
<tomcat-users>
  <user username="aadc" password="yourpass" roles="admin,manager">
</tomcat-users>
```

### 3.3. Webapps Directory

Tomcat for some reason created two webapps directories; one in `/var/lib/tomcat6` and one in `/usr/share/tomcat6`. The latter is not used, so you can rename it to `webapps.unused` so as not to get it confused with the other one.

```
# mv /usr/share/tomcat6/webapps /usr/share/tomcat6/webapps.unused
```

You should also create a softlink to the other webapps directory here:

```
# ln -s /var/lib/tomcat6/webapps /usr/share/tomcat6/webapps
```

If you set up the webapps directory well, you can easily enable and disable webapps and prevent some from starting with Tomcat. This is especially useful if you are installing a new version of a web app such as Geoserver and want to be able to revert back to an older version.

First you should stop tomcat:

```
# /etc/init.d/tomcat6 stop
```

Next, rename the webapps directory to `webapps-available`, and create a new directory called `webapps-enabled`. Then create a symbolic link called `webapps` to `webapps-enabled`. The following shows how to achieve this:

```
# cd /var/lib/tomcat6
# mv webapps webapps-available
# mkdir webapps-enabled
# ln -s webapps-enabled webapps
```

You should also make sure that the `tomcat6` user owns those directories:

```
# chown -R tomcat6 /var/lib/tomcat6/webapps
```



Now you need to create symbolic links in the `webapps-enabled` folder to whichever applications you would like Tomcat to start from `webapps-available`. At a minimum, the `ROOT` application should be softlinked:

```
# cd /var/lib/tomcat6/webapps-enabled
# ln -s ../webapps-available/ROOT
```

Now you can deploy multiple versions of webapps such as Geoserver to the `webapps-available` directory (i.e. `geoserver-1.7.7`, `geoserver-2.0.1`, etc) but have a softlink (called `geoserver` in this case) that points to the version that you want activated.

If Tomcat is configured with auto-deploy (it is by default), just adding or removing a softlink to the `webapps-enabled` directory will cause it to deploy/undeploy.

## 4. Restart Tomcat

Now everything should be set up, you can restart Tomcat:

```
# /etc/init.d/tomcat6 restart
```

## 5. Deploying Webapps

Webapps can now be deployed by dropping the WAR file into the `/var/lib/tomcat6/webapps` directory.

It will automatically extract to a folder with the same name as the WAR file, minus the `.war` extension.

What this is actually doing is extracting the application to the `webapps-enabled` directory, so after it has finished deploying, you should stop Tomcat and move it to the `webapps-available` directory, then creating a softlink from `webapps-enabled` to it.

*It's not completely necessary to stop Tomcat but if you leave it running with the war file still in the webapps directory it will redeploy the app there after you move it. You can move both at once, but if you just remove the war file alone it will cause the webapp to undeploy too.*

You can then also remove the WAR file. The following uses Geoserver as an example:

```
# /etc/init.d/tomcat6 stop
# cd /var/lib/tomcat6/webapps-enabled
# mv geoserver-2.0.1 ../webapps-available
# ln -s ../webapps-available/Geoserver-2.0.1 geoserver
# rm geoserver-2.0.1.war
# /etc/init.d/tomcat6 start
```

Now the application will be available at <http://server.aad.gov.au:8080/geoserver>



## 6. Apache Forwarding

Apache web server adds a layer of security to tomcat applications that are being served to the outside world. To have apache forward tomcat apps (i.e. from port 8080 to port 80) you will need to install both apache2 and mod-jk:

```
# aptitude install apache2 libapache2-mod-jk
```

Next you need to edit `/etc/apache2/mods-enabled/jk.load` and include two lines for each application that you want forwarded. It should also include the `LoadModule` line, ONLY if it already exists, such as:

```
LoadModule jk_module /usr/lib/apache2/modules/mod_jk.so
```

```
JkMount /geoserver/* ajp13
```

```
JkMount /geoserver ajp13
```

This will forward the geoserver application from tomcat on port 8080 to port 80.

