PLANETJ CORPORATION

Configuring and Optimizing Tomcat

* TOMCAT *

Tomcat Security and Optimization Document

© PlanetJ Corporation
1821 Kelson Place, Escondido CA 92029
Phone 760-432-0600 • Fax 760-432-0600

Table of Contents

Chapter

1

INTRODUCTION

Introduction to Tomcat:

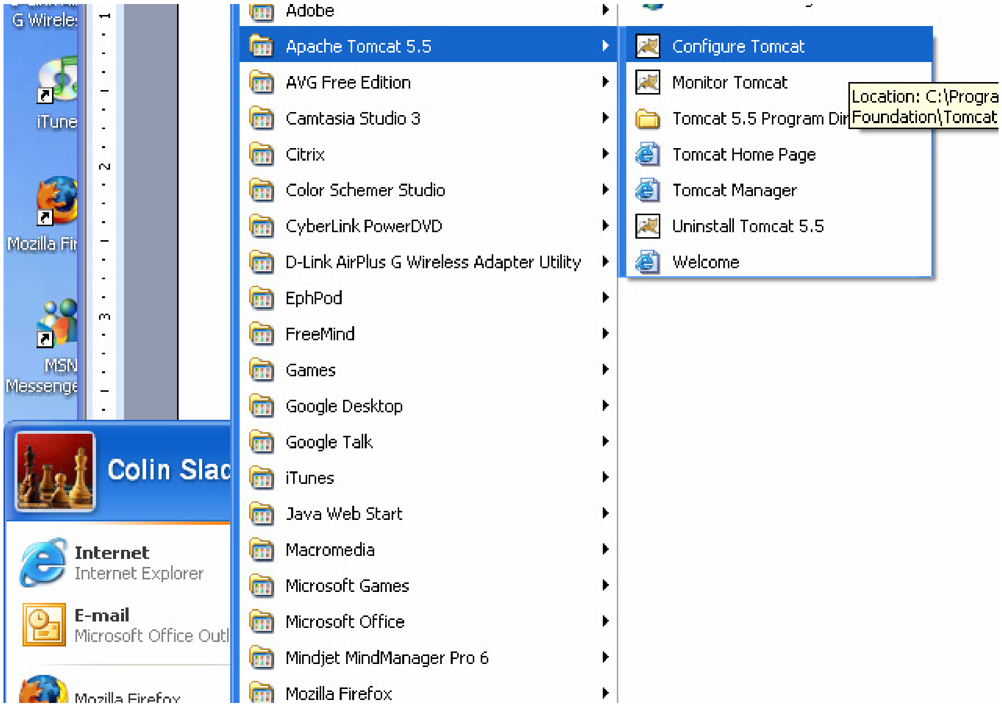
WebSphere has gained significant market share in Web application development space. A powerful, easy to use, and free alternative to WebSphere is Apache's Tomcat. Tomcat is open source and is the reference implementation for Sun' servlet standard. Tomcat has proven itself as production quality and is used by many companies. Tomcat installation on Window servers is an automated and easy process however the default installation places Tomcat into development mode. The advantage of development mode is that changes to JSPs, class, and JAR files are automatically incorporated at runtime without the need to restart the application server. The drawback is that performance suffers in development mode. Once you have created your WOW application and are ready to serve it to the public, the following steps should be taken to optimize Tomcat.

STEP 1

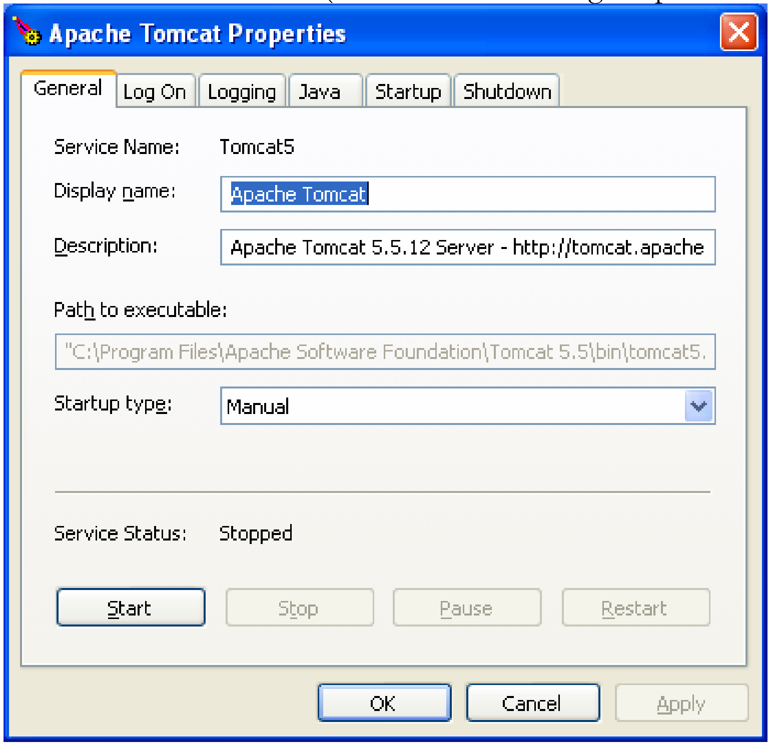
Increase Maximum Memory Used

The following recommendation does not apply to IBM iSeries because i5OS automatically manages memory. Tomcat ships with the maximum amount of memory set to 64MB, which is not large enough to operate most large web applications. We recommend increasing the maximum amount of memory to 80% of the total RAM available on your server. This can be done as shown below.

First, open Tomcat by going to the Start Menu-Programs-Apache Tomcat-Configure Tomcat.



After the Tomcat Configuration box has opened you should see the Service Status if stopped then click on the *Start button* to start Tomcat.(Recommended: Change all performance settings before starting Tomcat)



Service Name: Tomcat5

Display name: Apache Tomcat

Description: Apache Tomcat 5.5.12 Server - http://tomcat.apache

Path to executable:

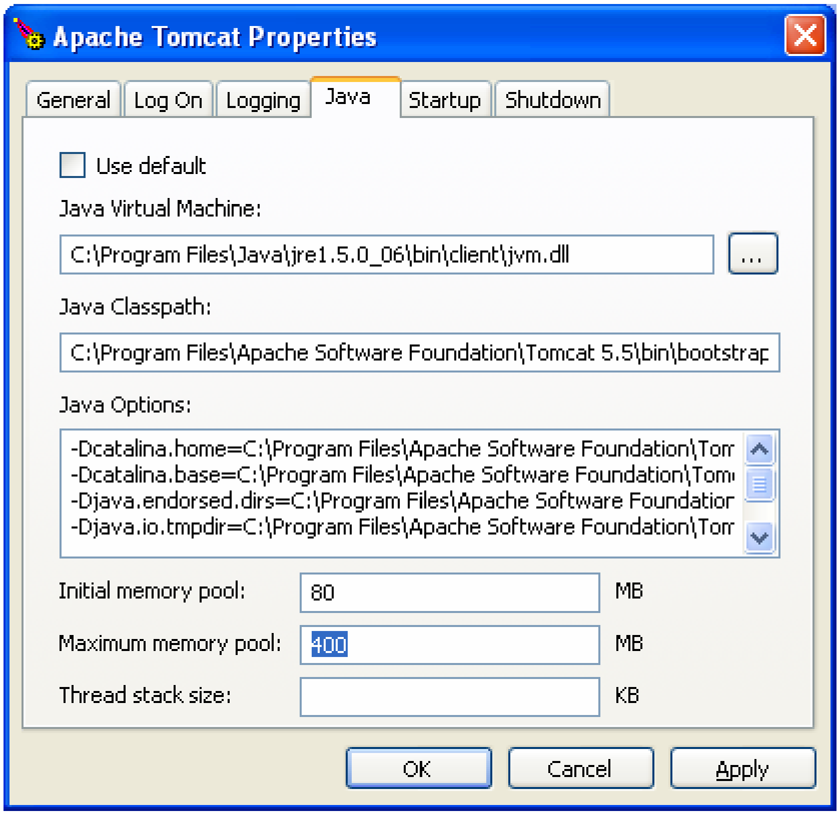
"C:\Program Files\Apache Software Foundation\Tomcat 5.5\bin\tomcat5.

Startup type: Manual

Service Status: Stopped

To change the Tomcat properties allowing changes to the maximum memory, click on the Java tab in the

Apache Tomcat Properties screen. You will see at the bottom of the window two dialog boxes *Initial memory pool* and *Maximum memory pool*. These are the initial Tomcat min and max memory allocations, which you will want to increase for large applications. (PlanetJ recommends that you increase memory after installing WOW to handle any future WOW Applications)



☐ Use default

Java Virtual Machine:
C:\Program Files\Java\jre1.5.0_06\bin\client\jvm.dll

Java Classpath:
C:\Program Files\Apache Software Foundation\Tomcat 5.5\bin\bootstrap

Java Options:

```
-Dcatalina.home=C:\Program Files\Apache Software Foundation\Tom
-Dcatalina.base=C:\Program Files\Apache Software Foundation\Tom
-Djava.endorsed.dirs=C:\Program Files\Apache Software Foundation
-Djava.io.tmpdir=C:\Program Files\Apache Software Foundation\Tom
```

Initial memory pool: 80 MB

Maximum memory pool: 400 MB

Thread stack size: KB

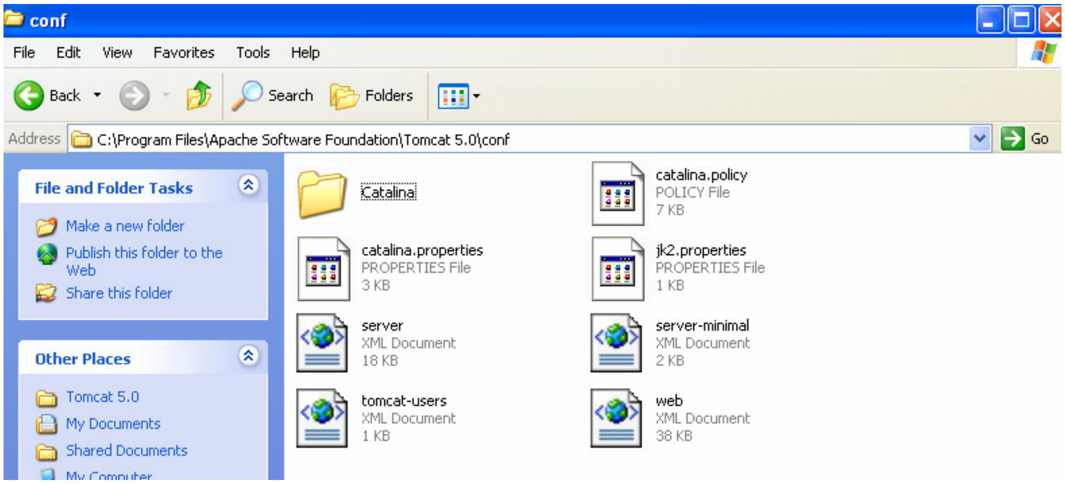
The minimum value can be anything but should start with at least 64-100MB. The maximum memory should be up to 80% of your computers RAM. For example, if you have 512MB of RAM then you should set the maximum memory to around 400MB. Enter in the new values and click on Apply

STEP 2

Changes to web.xml configuration file:

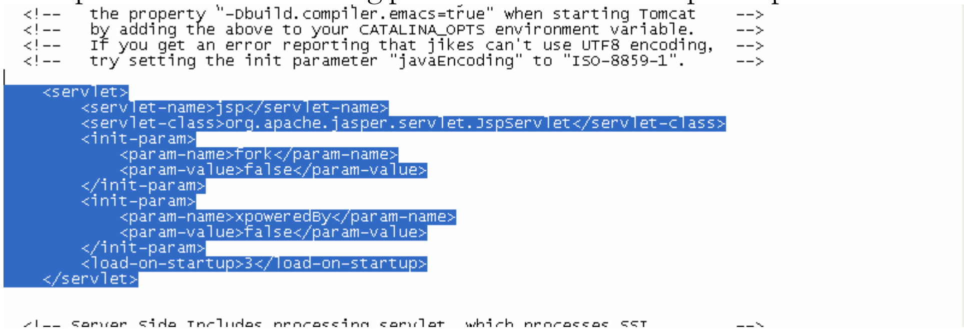
The web.xml is the document that defines default values for **all** web applications loaded into each instance of Tomcat. As each application is deployed, this file is processed, followed by the "/WEB-INF/web.xml" deployment descriptor for your own applications. The web.xml is located in Apache Software Foundation-

Tomcat-conf folder shown below.



C:\Program Files\Apache Software Foundation\Tomcat 5.0\conf

Open the web.xml by right clicking and select open with (NotePad or Wordpad or any text editor.) Now in the document we want to find the JSP Servlet Tag. This is located a quarter of the way down the page. Add two initial parameters - the reloading parameter and the development parameter.

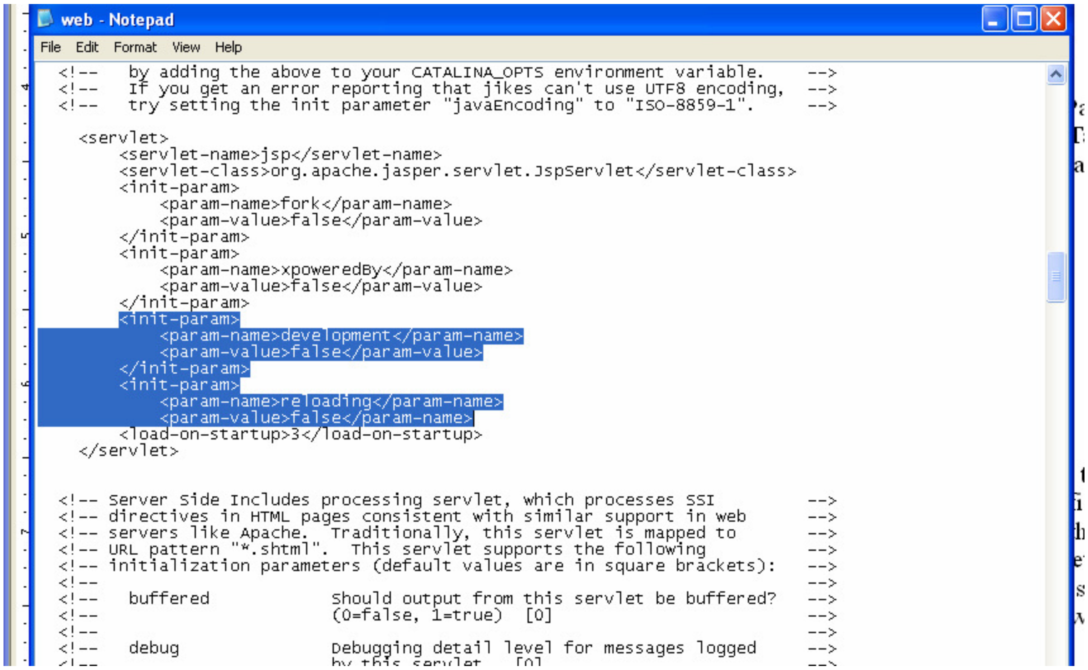


```
  <!--   the property "-Dbuild.compiler.emacs=true" when starting Tomcat    -->
  <!--   by adding the above to your CATALINA_OPTS environment variable.    -->
  <!--   If you get an error reporting that jikes can't use UTF8 encoding,  -->
  <!--   try setting the init parameter "javaEncoding" to "ISO-8859-1".     -->

    <servlet>
        <servlet-name>jsp</servlet-name>
        <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
        <init-param>
            <param-name>fork</param-name>
            <param-value>false</param-value>
        </init-param>
        <init-param>
            <param-name>xpoweredBy</param-name>
            <param-value>false</param-value>
        </init-param>
        <load-on-startup>3</load-on-startup>
    </servlet>


  <!-- Server Side Includes processing servlet, which processes SSI        -->
```

The reloading and development parameters are automatically set to true when Tomcat is installed. The development init-parameter will check for JSP modification on every access and reload. Set both of these parameters in the web.xml document to false so Tomcat will not look for or reload changes to JSPs on every access.



```
<!--   by adding the above to your CATALINA_OPTS environment variable.    -->
<!--   If you get an error reporting that jikes can't use UTF8 encoding,  -->
<!--   try setting the init parameter "javaEncoding" to "ISO-8859-1".     -->

  <servlet>
      <servlet-name>jsp</servlet-name>
      <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
      <init-param>
          <param-name>fork</param-name>
          <param-value>false</param-value>
      </init-param>
      <init-param>
          <param-name>xpoweredBy</param-name>
          <param-value>false</param-value>
      </init-param>
      <init-param>
          <param-name>development</param-name>
          <param-value>false</param-value>
      </init-param>
      <init-param>
          <param-name>reloading</param-name>
          <param-value>false</param-name>
      <load-on-startup>3</load-on-startup>
  </servlet>


<!-- Server Side Includes processing servlet, which processes SSI         -->
<!-- directives in HTML pages consistent with similar support in web      -->
<!-- servers like Apache.  Traditionally, this servlet is mapped to       -->
<!-- URL pattern "*.shtml".  This servlet supports the following          -->
<!-- initialization parameters (default values are in square brackets):   -->
<!--                                                                      -->
<!--   buffered            Should output from this servlet be buffered?   -->
<!--                       (0=false, 1=true)  [0]                         -->
<!--                                                                      -->
<!--   debug               Debugging detail level for messages logged     -->
<!--                       by this servlet    [0]                         -->
```

To add the init-parameters you will add this code:

```
<init-param>
        <param-name>development</param-name>
           <param-value>false</param-value>
</init-param>
<init-param>
        <param-name>reloading</param-name>
        <param-value>false</param-value>
</init-param>
```

In summary, Tomcat is a cost-efficient tool to hosting Web applications. Now that you have made the above simple changes, Tomcat will run much faster and handle more complex workloads.

STEP 3

Disable Session Persistence

WOW does not support session persistence and it should be disabled to avoid unpredictable results after server restarts. Session persistence is intended to allow browser interactions to be "saved and restored" after the application server is stopped and restarted. Due to WOW's RAM based magic request architecture, session persistence is not compatible.

To disable session persistence uncomment the Manager tag that is in:

<tomcat-home>/conf/context..xml

```
  <!--     Uncomment     this     to     disable     session     persistence     across     Tomcat     restarts     -->
  <!--
  <Manager                                         pathname=""                                                  />
  -->
```

To uncomment remove the comments which are matching <!-- this is a comment --> change the text to appear as follows:

```
<!--     Uncomment     this     to     disable     session     persistence     across     Tomcat     restarts     -->

  <Manager                                         pathname=""                                                  />
```

With the comments removed this configuration file which cause a "blank" session manager to be assign thus stopping session persistence.

STEP 4

Increase Session Timeout (Optional)

WOW defaults to keeping a session active for 4 hours (240 minutes) before it is "timed out" to conserve memory. Timeout can be an inconvenience to users who access WOW applications in a frequent manner. For frequent users, you may increase the timeout to 36 hours by modifying the file located at:

<tomcat-home>/webapps/wowXX/web-inf/web..xml

NOTE: Replace XX with your WOW version number. A restart of the Application Server is required to changes to take effect.

Near the bottom of this file, the Session timeout is declared. Carefully change 240 to 2160 or some other desire value. This is the time out in minutes.

```
<!-- Below is the Session Timeout-->

     <session-config id="Session_Timeout">

        <session-timeout>2160</session-timeout>
```

```
</session-config>
```

Chapter

2

TOMCAT ADMINISTRATOR AND MANAGER APPLICATIONS

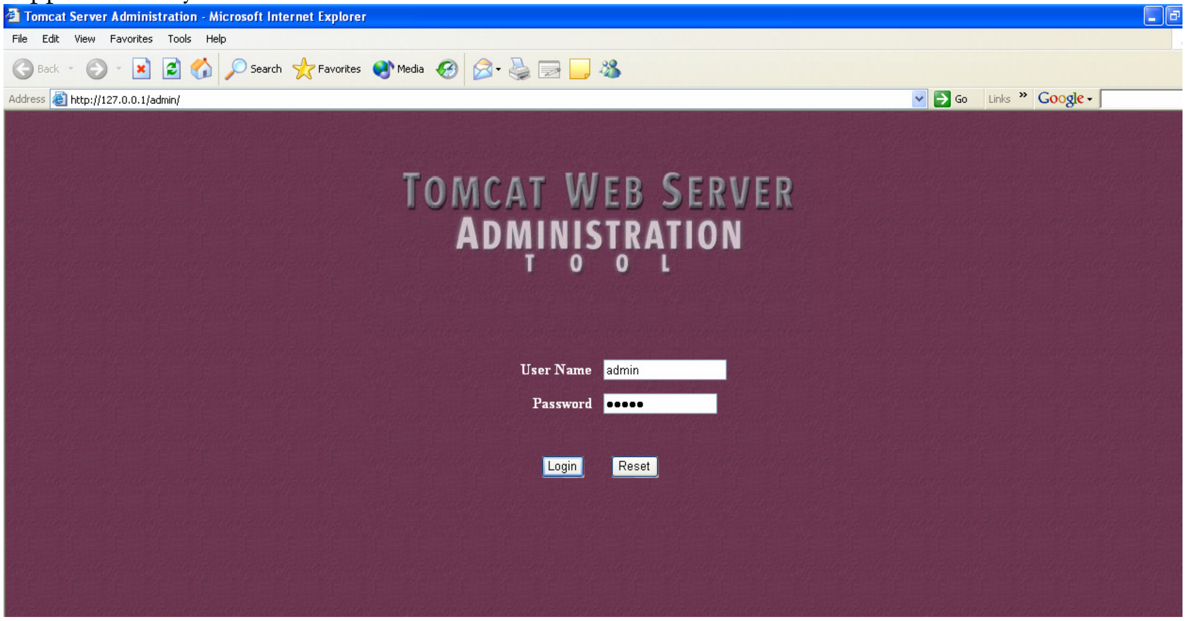
Administering Tomcat:

Tomcat also has its own administration, which can be helpful and accessed through the Tomcat server. Now we are going to show some of the helpful and powerful administration applications that Tomcat ships with such as Tomcat Administrator and Tomcat Manager.

To access both of these you can either go through the start menu under Apache Tomcat folder or you can open a browser and type go to the localhost such as http://localhost:80 with the port that you specified in the installation of Tomcat.

NOTE: Tomcat Administrator is not installed by Default. If needed you must run "Admin" package.

Tomcat allows administration (start, stop, destroy, manage, create etc) via a web application. The Tomcat Administration Application allows you to see the server information, edit connections, edit users, groups, resources and change the main settings of the webserver. This application runs using the userid and password supplied when you installed Tomcat.

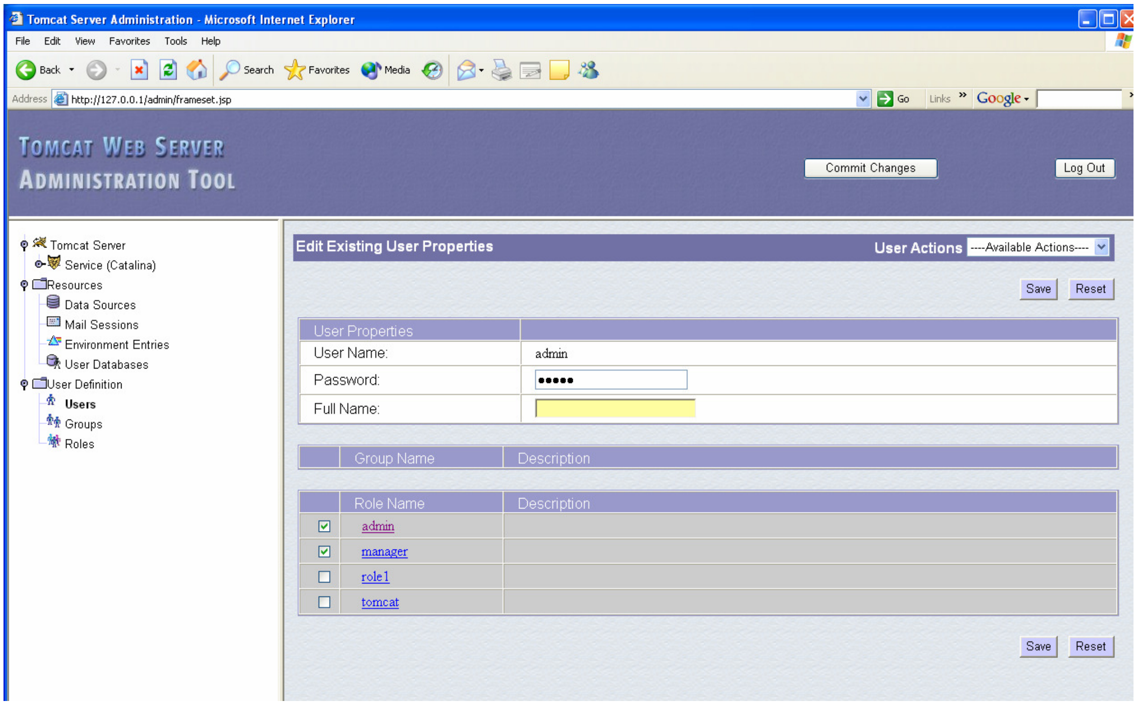


Tomcat Server Administration - Microsoft Internet Explorer

Address: http://127.0.0.1/admin/

User Name: admin

Password: •••••



User Properties	
User Name:	admin
Password:	•••••
Full Name:	

	Group Name	Description

	Role Name	Description
☑	admin	
☑	manager	
☐	role1	
☐	tomcat	

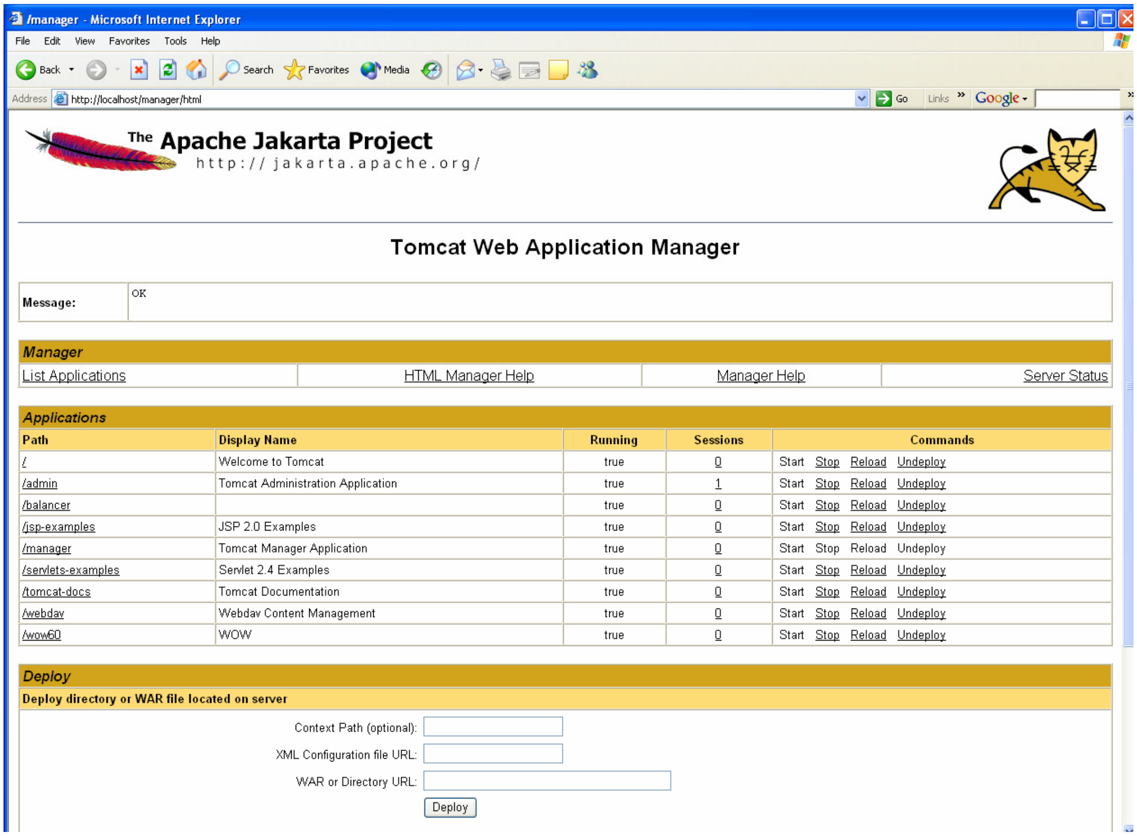
The userid and pwd are defined in tomcat at:

..../config/tomcat_users.xml as shown below.

```
<?xml version=”1.0” encoding=”utf-8” ?>
- <tomcat-users>
<role rolename=”tomcat” />
<role rolename=”role1” />
<role rolename=”manager” />
<role rolename=”admin” />
<user username=”tomcat” password=”tomcat” roles=”tomcat” />
<user username=”role1” password=”tomcat” roles=”role1” />
<user username=”both” password=”tomcat” roles=”tomcat,role1” />
<user username=”admin” password=”admin” roles=”admin,manager” />
</tomcat-user>
```

These are set when you first install Tomcat on to your machine when the installation asks for username and password. You can also change the username and password and add new users in Tomcat Administration Application.

Tomcat also ships with the Tomcat Manager. This is an application that will allow you to edit, start, and stop applications dynamically without restarting the Tomcat. It also shows current applications and a few other options, which you can learn about by clicking on the Manager Help Link Shown below.



/manager - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Go Links Google

Address http://localhost/manager/html

The Apache Jakarta Project
http://jakarta.apache.org/

Tomcat Web Application Manager

Message:	OK

Manager

List Applications	HTML Manager Help	Manager Help	Server Status

Applications

Path	Display Name	Running	Sessions	Commands
/	Welcome to Tomcat	true	0	Start Stop Reload Undeploy
/admin	Tomcat Administration Application	true	1	Start Stop Reload Undeploy
/balancer		true	0	Start Stop Reload Undeploy
/jsp-examples	JSP 2.0 Examples	true	0	Start Stop Reload Undeploy
/manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy
/servlets-examples	Servlet 2.4 Examples	true	0	Start Stop Reload Undeploy
/tomcat-docs	Tomcat Documentation	true	0	Start Stop Reload Undeploy
/webdav	Webdav Content Management	true	0	Start Stop Reload Undeploy
/wow60	WOW	true	0	Start Stop Reload Undeploy

Deploy

Deploy directory or WAR file located on server

Context Path (optional):

XML Configuration file URL:

WAR or Directory URL:

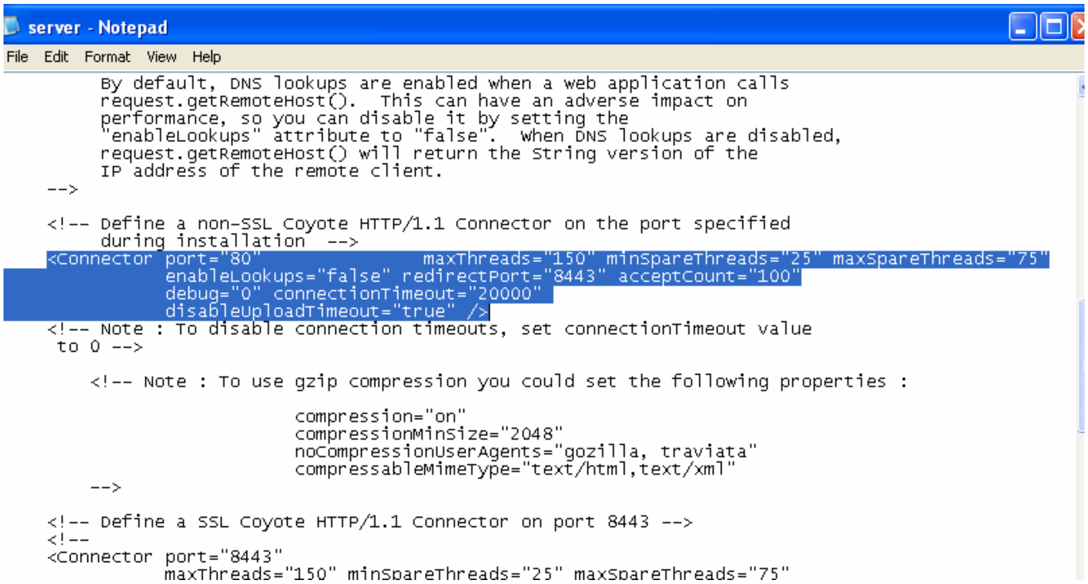
Chapter

3

CHANGING TOMCAT PORT SETTINGS

Changing Port Settings:

When you install Apache Tomcat it defaults the port to 8080. The ports are set in the ..Tomcat/conf/server.xml document. To adjust the port being used right click and open with notepad or any other text editor. After the document is open scroll down around half way looking for the connecter tag shown below and change the port number to a new port number (Usually either 8080 or 80). Then save the newly changed server.xml file and restart Tomcat.



```
        By default, DNS lookups are enabled when a web application calls
        request.getRemoteHost().  This can have an adverse impact on
        performance, so you can disable it by setting the
        "enableLookups" attribute to "false".  When DNS lookups are disabled,
        request.getRemoteHost() will return the String version of the
        IP address of the remote client.
    -->

    <!-- Define a non-SSL Coyote HTTP/1.1 Connector on the port specified
         during installation  -->
    <Connector port="80"               maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
               enableLookups="false" redirectPort="8443" acceptCount="100"
               debug="0" connectionTimeout="20000"
               disableUploadTimeout="true" />
    <!-- Note : To disable connection timeouts, set connectionTimeout value
     to 0 -->

        <!-- Note : To use gzip compression you could set the following properties :

                           compression="on"
                           compressionMinSize="2048"
                           noCompressionUserAgents="gozilla, traviata"
                           compressableMimeType="text/html,text/xml"
        -->

    <!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
    <!--
    <Connector port="8443"
               maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
```

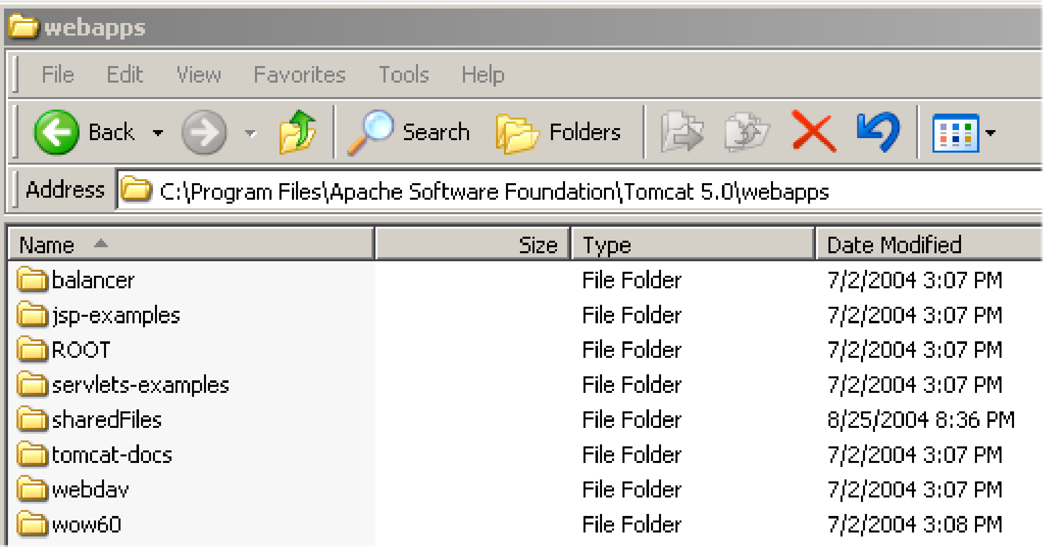
Chapter

4

SECURING APPLICATIONS FILES IN TOMCAT

Secure Application Files in Tomcat:

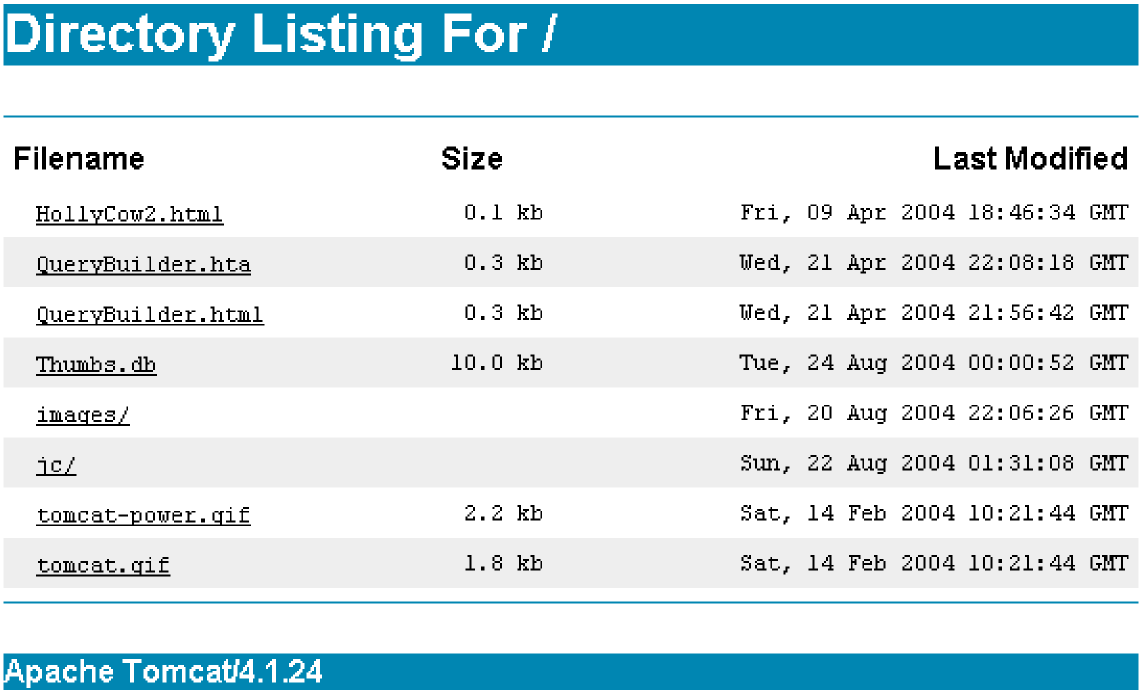
When installing a new web application such as WOW to Tomcat, Tomcat automatically allows people to view the directory listing of all files inside of the webapps folder. An example of this is the ***sharedFiles*** folder in the webapps folder on the hard drive shown below.



C:\Program Files\Apache Software Foundation\Tomcat 5.0\webapps

Name	Size	Type	Date Modified
balancer		File Folder	7/2/2004 3:07 PM
jsp-examples		File Folder	7/2/2004 3:07 PM
ROOT		File Folder	7/2/2004 3:07 PM
servlets-examples		File Folder	7/2/2004 3:07 PM
sharedFiles		File Folder	8/25/2004 8:36 PM
tomcat-docs		File Folder	7/2/2004 3:07 PM
webdav		File Folder	7/2/2004 3:07 PM
wow60		File Folder	7/2/2004 3:08 PM

By Default the ***sharedFiles*** folder includes three directories: one directory contains sample images and another directory contains sample MS Word Documents. These folders are for example purpose only and may be deleted or added upon. There is also a directory labeled ***WEB-INF***. This folder contains a file called web.xml. Web.xml allows WOW to be set up to show its contents over the Internet or to be set so that those files cannot be seen over the Internet. Below is an example of connecting to the ***sharedFiles*** directory through Tomcat.

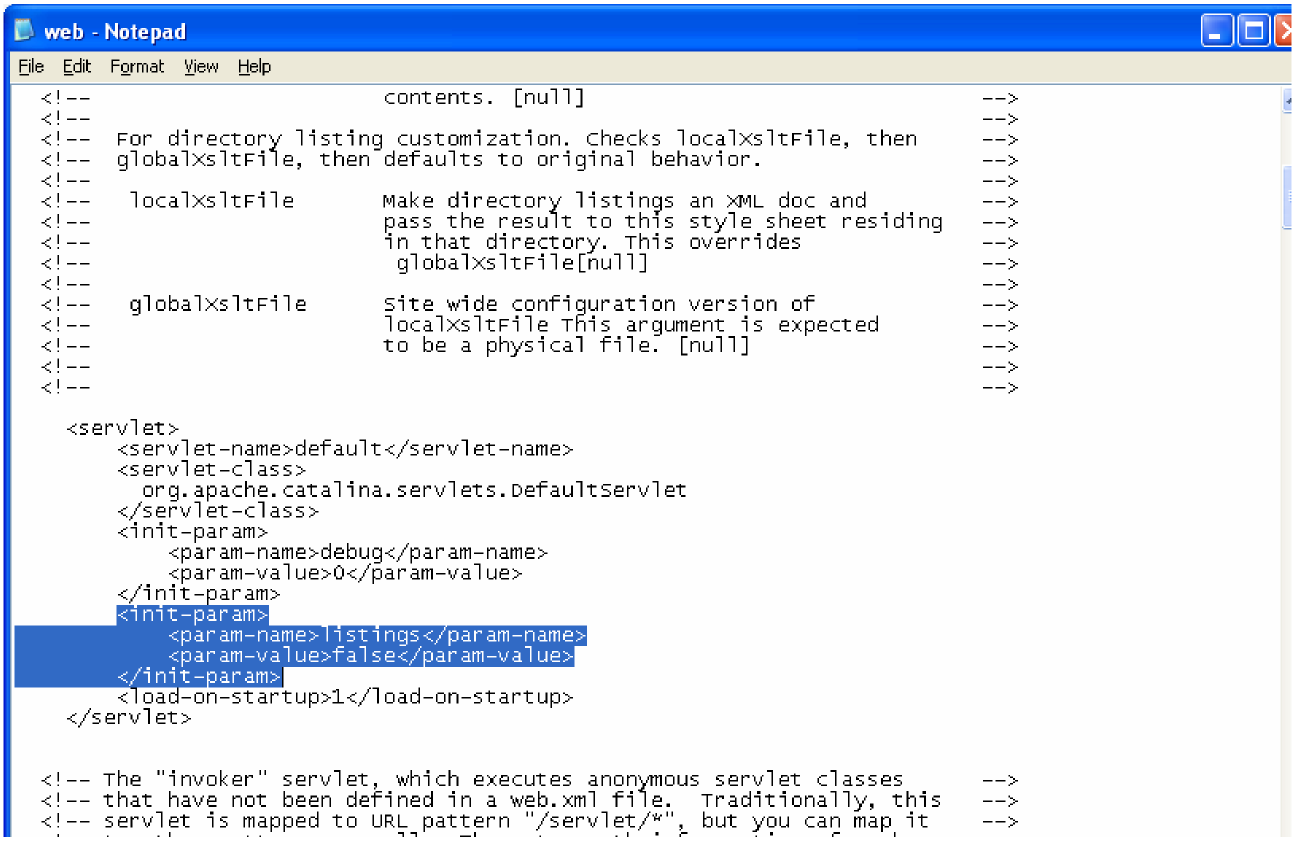


Directory Listing For /

Filename	Size	Last Modified
HollyCow2.html	0.1 kb	Fri, 09 Apr 2004 18:46:34 GMT
QueryBuilder.hta	0.3 kb	Wed, 21 Apr 2004 22:08:18 GMT
QueryBuilder.html	0.3 kb	Wed, 21 Apr 2004 21:56:42 GMT
Thumbs.db	10.0 kb	Tue, 24 Aug 2004 00:00:52 GMT
images/		Fri, 20 Aug 2004 22:06:26 GMT
jc/		Sun, 22 Aug 2004 01:31:08 GMT
tomcat-power.gif	2.2 kb	Sat, 14 Feb 2004 10:21:44 GMT
tomcat.gif	1.8 kb	Sat, 14 Feb 2004 10:21:44 GMT

Apache Tomcat/4.1.24

This creates a security problem, as the whole Internet will be able to see our code and mess around with our files. To disable this feature you need to go in Apache Tomcat file structure <Apache Tomcat>/conf/web.xml file. In this web.xml file shown below there is an initial parameter called listing that is automatically set to true. We want to change that to false so that your files cannot be directly accessed and seen over the Internet.



```
<!--                       contents.  [null]                              -->
<!--                                                                      -->
<!--  For directory listing customization. Checks localXsltFile, then     -->
<!--  globalXsltFile, then defaults to original behavior.                 -->
<!--                                                                      -->
<!--   localXsltFile       Make directory listings an XML doc and         -->
<!--                       pass the result to this style sheet residing   -->
<!--                       in that directory. This overrides              -->
<!--                        globalXsltFile[null]                          -->
<!--                                                                      -->
<!--   globalXsltFile      Site wide configuration version of             -->
<!--                       localXsltFile This argument is expected        -->
<!--                       to be a physical file.  [null]                 -->
<!--                                                                      -->
<!--                                                                      -->

  <servlet>
      <servlet-name>default</servlet-name>
      <servlet-class>
        org.apache.catalina.servlets.DefaultServlet
      </servlet-class>
      <init-param>
          <param-name>debug</param-name>
          <param-value>0</param-value>
      </init-param>
      <init-param>
          <param-name>listings</param-name>
          <param-value>false</param-value>
      </init-param>
      <load-on-startup>1</load-on-startup>
  </servlet>


<!-- The "invoker" servlet, which executes anonymous servlet classes      -->
<!-- that have not been defined in a web.xml file.  Traditionally, this   -->
<!-- servlet is mapped to URL pattern "/servlet/*", but you can map it    -->
```

Now that listings has been set to false users will not be able to see our files by connecting through Tomcat.

Chapter

5

TOMCAT SECURITY AND SSL

SSL:

To set up SSL and other security features for Tomcat refer to the Apache's Tomcat site for SSL setup and optimization, http://jakarta.apache.org/tomcat

Other Security Considerations:

1. Make sure your have set proper userids and passwords for your Tomcat user file. This file is located at: ..../config/tomcat_users.xml

References

Tomcat: http://jakarta.apache.org/tomcat
PlanetJ's WOW: http://www.planetjavainc.com