Config – 2

Agenda: Once Tomcat is up and running, you will want to keep an eye on it, to help it along occasionally.

* Various places to look for information about your server,
* Troubleshooting how to find out why things aren’t working
* Common mistakes in setting up and configuring Tomcat.
* How and why to run Tomcat on the default HTTP port 80?
* Pointers on what JVM start-up switch settings manage the web user accounts that Tomcat knows about
* Configuring security realms to customize which users can access your Tomcat’s web content.
* Configuring Tomcat to open a pool of connections to your database for your webapp to use.
* Configuring Tomcat to use Common Gateway Interface (CGI) programs.
* Tomcat administration web application, a tool for helping you with the task of keeping Tomcat running.

Item1: you should not store your web application’s files in the Tomcat distribution’s directory tree.

Because

* When you install a newer version of Tomcat’s files into the Tomcat installation directory tree, you may overwrite the server.xml and any other configuration files that you modified for your site/application.
* Usually, package managers save the file they’re replacing if it is a known configuration file, but even then it’s a pain to know what you need to do to get your site back in running order.

So it is a good idea to keep your web site’s files cleanly separate from the Tomcat distribution files.

Another scenario

You install one copy of the Tomcat distribution, but you wish to run more than one instance of Tomcat on your server computer.

reasons : having each one serve different content on different TCP ports and you want each webapp in its own JVM so they can be operated independently.

You must keep each JVM instance’s files separate.

Normally Tomcat(the server) reads configuration from the conf and webapps directories and writes files to the logs, temp, and work directories.

Also, some jar files and class files may need to be loaded from the shared, server, and common directory trees.

This means that for multiple instances to work, each Tomcat instance has to have its own set of these directories; they cannot be shared by two differently configured Tomcat JVM instances.

The making this work you must set the CATALINA\_HOME environment variable to where you installed the Tomcat binary distribution, and

You must set the CATALINA\_BASE environment variable to a different path where you are storing a JVM instance’s files (these files come from you).

When you have both of these environment variables set and you start Tomcat, it will run using your files in CATALINA\_BASE, on top of the Tomcat binary distribution in CATALINA\_HOME.

This is built-in feature of Tomcat allows you to keep Tomcat’s files separate from your files, but still makes it possible to modify everything you need to modify to configure everything the way you need it to be.

<https://stackoverflow.com/questions/4678077/can-i-run-two-tomcat-instances-on-the-same-machine>

Let's say that you have only one Tomcat folder located in **C:\apache-tomcat-7.0.39**, and that you wish to run two instances from it.

* Set CATALINA\_HOME system/user variable pointing to C:\apache-tomcat-7.0.39
* Create a folder C:\instance1.
  + Copy conf, webapps and temp folders from C:\apache-tomcat-7.0.39.
  + Delete contents from webapps and temp folders located under instance1,
  + But don't touch conf contents.
* Now copy>paste C:\instance1 and rename it to instance2.
  + That way, both instance1 and instance2 will have the same content.
* Go to C:\instance2\conf, edit server.xml and change the numbers of these ports.

Deploy whatever you want into instance1\webapps and instance2\webapps Create the following 4 batch files under C:\

Set CATALINA\_HOME= C:\apache-tomcat-7.0.39

**instance1\_startup.bat**

@echo off

set CATALINA\_BASE=C:\instance1

cd "%CATALINA\_HOME%\bin"

set TITLE=My Tomcat Instance 01

call startup.bat %TITLE%

**instance1\_shutdown.bat**

@echo off

set CATALINA\_BASE=C:\instance1

call %CATALINA\_HOME%\bin \shutdown.bat

**instance2\_startup.bat**

@echo off

set CATALINA\_BASE=C:\instance2

set TITLE=My Tomcat Instance 02

call %CATALINA\_HOME%\bin\startup.bat %TITLE%

**instance2\_shutdown.bat**

@echo off

set CATALINA\_BASE=C:\instance2

cd "%CATALINA\_HOME%\bin"

call shutdown.bat

Run instance1\_startup.bat and instance2\_startup.bat, hopefully it should work.

Tomcat installation directories:

<https://www.ntu.edu.sg/home/ehchua/programming/howto/Tomcat_More.html>

* bin: for Tomcat's binaries and start-up scripts.
* conf: global configuration applicable to all the webapps. The default installation provides:
  + Policy File:1 - catalina.policy for specifying security policy.
  + Properties Files:2 - catalina.properties and logging.properties,
  + Configuration XML Files:4 –
    - server.xml: Tomcat main configuration file,
    - web.xml : global web application deployment descriptors,
    - context.xml : global Tomcat-specific configuration options
    - tomcat-users.xml: user, password and role for authentication and access control.
  + The conf also contain a sub-directory for each engine, e.g., Catalina, which in turn contains a sub-sub-directory for each of its hosts, e.g., localhost. You can place the host-specific context information (similar to context.xml, but named as webapp.xml for each webapp under the host).
* lib: Keeps the JAR-file that are available to all webapps. The default installation include
  + Jars which make up tomcat : Catalina.jar, tomcat-api.jar
  + Servlet support jars : servlet-api.jar (Servlet), jasper.jar (JSP) and jasper-el.jar (EL).
  + You may also keep the JAR files of external package here: such as MySQL JDBC driver (mysql-connector-java-5.1.{xx}-bin.jar) and JSTL (jstl.jar and standard.jar).
* logs: contains the engine logfile Catalina.{yyyy-mm-dd}.log, host logfile localhost.{yyyy-mm-dd}.log, and other application logfiles such as manger and host-manager. The access log (created by the AccessLogValve) is also kept here.
* webapps: the default appBase - web applications base directory of the host localhost.
* work: contains the translated servlet source files and classes of JSP/JSF. Organized in hierarchy of engine name (Catalina), host name (localhost), webapp name, followed by the Java classes package structure.
* temp: temporary files.