Running The Apache Tomcat 9.0 Servlet/JSP Container

Apache Tomcat 9.0 requires a Java Standard Edition Runtime
Environment (JRE) version 8 or later.

Running With JRE 8 Or Later

(1) Download and Install a Java SE Runtime Environment (JRE)

(1.1) Download a Java SE Runtime Environment (JRE),
      release version 8 or later, from
      http://www.oracle.com/technetwork/java/javase/downloads/index.html

(1.2) Install the JRE according to the instructions included with the
      release.

      You may also use a full Java Development Kit (JDK) rather than just
      a JRE.

(2) Download and Install Apache Tomcat

(2.1) Download a binary distribution of Tomcat from:

      https://tomcat.apache.org/

(2.2) Unpack the binary distribution so that it resides in its own
      directory (conventionally named "apache-tomcat-[version]").

      For the purposes of the remainder of this document, the name
      "CATALINA_HOME" is used to refer to the full pathname of that
      directory.

NOTE: As an alternative to downloading a binary distribution, you can
create your own from the Tomcat source code, as described in
"BUILDING.txt". You can either

  a) Do the full "release" build and find the created distribution in the
      "output/release" directory and then proceed with unpacking as above, or

  b) Do a simple build and use the "output/build" directory as
      "CATALINA_HOME". Be warned that there are some differences between the
      contents of the "output/build" directory and a full "release"
      distribution.

(3) Configure Environment Variables

Tomcat is a Java application and does not use environment variables directly. Environment variables are used by the Tomcat startup scripts. The scripts use the environment variables to prepare the command that starts Tomcat.

(3.1) Set CATALINA_HOME (required) and CATALINA_BASE (optional)

The CATALINA_HOME environment variable should be set to the location of the root directory of the "binary" distribution of Tomcat.

The Tomcat startup scripts have some logic to set this variable automatically if it is absent, based on the location of the startup script in *nix and on the current directory in Windows. That logic might not work in all circumstances, so setting the variable explicitly is recommended.

The CATALINA_BASE environment variable specifies location of the root directory of the "active configuration" of Tomcat. It is optional. It defaults to be equal to CATALINA_HOME.

Using distinct values for the CATALINA_HOME and CATALINA_BASE variables is recommended to simplify further upgrades and maintenance. It is documented in the "Multiple Tomcat Instances" section below.

(3.2) Set JRE_HOME or JAVA_HOME (required)

These variables are used to specify location of a Java Runtime Environment or of a Java Development Kit that is used to start Tomcat.

The JRE_HOME variable is used to specify location of a JRE. The JAVA_HOME variable is used to specify location of a JDK.

Using JAVA_HOME provides access to certain additional startup options that are not allowed when JRE_HOME is used.

If both JRE_HOME and JAVA_HOME are specified, JRE_HOME is used.

The recommended place to specify these variables is a "setenv" script. See below.

(3.3) Other variables (optional)

Other environment variables exist, besides the four described above. See the comments at the top of catalina.bat or catalina.sh scripts for the list and a description of each of them.

One frequently used variable is CATALINA_OPTS. It allows specification of additional options for the java command that starts Tomcat.

See the Java documentation for the options that affect the Java Runtime Environment.

See the "System Properties" page in the Tomcat Configuration Reference for the system properties that are specific to Tomcat.

A similar variable is JAVA_OPTS. It is used less frequently. It allows specification of options that are used both to start and to stop Tomcat as well as for other commands.

Note: Do not use JAVA_OPTS to specify memory limits. You do not need much memory for a small process that is used to stop Tomcat. Those settings belong to CATALINA_OPTS.

Another frequently used variable is CATALINA_PID (on *nix only). It specifies the location of the file where process id of the forked Tomcat java process will be written. This setting is optional. It will enable the following features:

 * better protection against duplicate start attempts and
 * allows forceful termination of Tomcat process when it does not react to
    the standard shutdown command.

(3.4) Using the "setenv" script (optional, recommended)

Apart from CATALINA_HOME and CATALINA_BASE, all environment variables can
be specified in the "setenv" script. The script is placed either into
CATALINA_BASE/bin or into CATALINA_HOME/bin directory and is named
setenv.bat (on Windows) or setenv.sh (on *nix). The file has to be
readable.

By default the setenv script file is absent. If the script file is present
both in CATALINA_BASE and in CATALINA_HOME, the one in CATALINA_BASE is
preferred.

For example, to configure the JRE_HOME and CATALINA_PID variables you can
create the following script file:

On Windows, %CATALINA_BASE%\bin\setenv.bat:

  set "JRE_HOME=%ProgramFiles%\Java\jre8"
  exit /b 0

On *nix, $CATALINA_BASE/bin/setenv.sh:

  JRE_HOME=/usr/java/latest
  CATALINA_PID="$CATALINA_BASE/tomcat.pid"

The CATALINA_HOME and CATALINA_BASE variables cannot be configured in the
setenv script, because they are used to locate that file.

All the environment variables described here and the "setenv" script are
used only if you use the standard scripts to launch Tomcat. For example, if
you have installed Tomcat as a service on Windows, the service wrapper
launches Java directly and does not use the script files.

(4) Start Up Tomcat

(4.1) Tomcat can be started by executing one of the following commands:

  On Windows:

      %CATALINA_HOME%\bin\startup.bat

    or

      %CATALINA_HOME%\bin\catalina.bat start

  On *nix:

      $CATALINA_HOME/bin/startup.sh

    or

      $CATALINA_HOME/bin/catalina.sh start

(4.2) After startup, the default web applications included with Tomcat will be
      available by visiting:

      http://localhost:8080/

(4.3) Further information about configuring and running Tomcat can be found in
      the documentation included here, as well as on the Tomcat web site:

      https://tomcat.apache.org/

(5) Shut Down Tomcat

(5.1) Tomcat can be shut down by executing one of the following commands:

  On Windows:

      %CATALINA_HOME%\bin\shutdown.bat

    or

      %CATALINA_HOME%\bin\catalina.bat stop

  On *nix:

      $CATALINA_HOME/bin/shutdown.sh

    or

      $CATALINA_HOME/bin/catalina.sh stop

==
Advanced Configuration - Multiple Tomcat Instances
==

In many circumstances, it is desirable to have a single copy of a Tomcat
binary distribution shared among multiple users on the same server. To make
this possible, you can set the CATALINA_BASE environment variable to the
directory that contains the files for your 'personal' Tomcat instance.

When running with a separate CATALINA_HOME and CATALINA_BASE, the files
and directories are split as following:

In CATALINA_BASE:

 * bin - Only the following files:

           * setenv.sh (*nix) or setenv.bat (Windows),
           * tomcat-juli.jar

          The setenv scripts were described above. The tomcat-juli library
          is documented in the Logging chapter in the User Guide.

 * conf - Server configuration files (including server.xml)

 * lib - Libraries and classes, as explained below

 * logs - Log and output files

 * webapps - Automatically loaded web applications

 * work - Temporary working directories for web applications

 * temp - Directory used by the JVM for temporary files (java.io.tmpdir)

In CATALINA_HOME:

 * bin - Startup and shutdown scripts

          The following files will be used only if they are absent in
          CATALINA_BASE/bin:

          setenv.sh (*nix), setenv.bat (Windows), tomcat-juli.jar

 * lib - Libraries and classes, as explained below

 * endorsed - Libraries that override standard "Endorsed Standards"

              libraries provided by JRE. See Classloading documentation
              in the User Guide for details.
              This is only supported for Java <= 8.
              By default this "endorsed" directory is absent.

In the default configuration the JAR libraries and classes both in
CATALINA_BASE/lib and in CATALINA_HOME/lib will be added to the common
classpath, but the ones in CATALINA_BASE will be added first and thus will
be searched first.

The idea is that you may leave the standard Tomcat libraries in
CATALINA_HOME/lib and add other ones such as database drivers into
CATALINA_BASE/lib.

In general it is advised to never share libraries between web applications,
but put them into WEB-INF/lib directories inside the applications. See
Classloading documentation in the User Guide for details.

It might be useful to note that the values of CATALINA_HOME and
CATALINA_BASE can be referenced in the XML configuration files processed
by Tomcat as ${catalina.home} and ${catalina.base} respectively.

For example, the standard manager web application can be kept in
CATALINA_HOME/webapps/manager and loaded into CATALINA_BASE by copying
its deployment descriptor into the desired virtual host:

 * Copy the CATALINA_HOME/webapps/manager/META-INF/context.xml
   file as CATALINA_BASE/conf/Catalina/localhost/manager.xml

 * Add docBase attribute as shown below.

The file will look like the following:

  <?xml version="1.0" encoding="UTF-8"?>
  <Context docBase="${catalina.home}/webapps/manager"
    antiResourceLocking="false" privileged="true" >
    <Valve className="org.apache.catalina.valves.RemoteAddrValve"
         allow="127\.0\.0\.1" />
    <Manager sessionAttributeValueClassNameFilter="java\.lang\.
(?:Boolean|Integer|Long|Number|String)|org\.apache\.catalina\.filters\.CsrfPreventionFilter\$LruCach
e(?:\$1)?|java\.util\.(?:Linked)?HashMap"/>
  </Context>

See Deployer chapter in User Guide and Context and Host chapters in the
Configuration Reference for more information on contexts and web
application deployment.

================
Troubleshooting
================

There are only really 2 things likely to go wrong during the stand-alone
Tomcat install:

(1) The most common hiccup is when another web server (or any process for that
    matter) has laid claim to port 8080. This is the default HTTP port that
    Tomcat attempts to bind to at startup. To change this, open the file:

       $CATALINA_HOME/conf/server.xml

    and search for '8080'. Change it to a port that isn't in use, and is
    greater than 1024, as ports less than or equal to 1024 require superuser
    access to bind under UNIX.

    Restart Tomcat and you're in business. Be sure that you replace the "8080"
    in the URL you're using to access Tomcat. For example, if you change the
    port to 1977, you would request the URL http://localhost:1977/ in your

browser.

(2) The 'localhost' machine isn't found. This could happen if you're behind a
    proxy. If that's the case, make sure the proxy configuration for your
    browser knows that you shouldn't be going through the proxy to access the
    "localhost".

    In Firefox, this is under Tools/Preferences -> Advanced/Network ->
    Connection -> Settings..., and in Internet Explorer it is Tools ->
    Internet Options -> Connections -> LAN Settings.

====================
Optional Components
====================

The following optional components may be included with the Apache Tomcat binary
distribution. If they are not included, you can install them separately.

 1. Apache Tomcat Native library

 2. Apache Commons Daemon service launcher

Both of them are implemented in C language and as such have to be compiled
into binary code. The binary code will be specific for a platform and CPU
architecture and it must match the Java Runtime Environment executables
that will be used to launch Tomcat.

The Windows-specific binary distributions of Apache Tomcat include binary
files for these components. On other platforms you would have to look for
binary versions elsewhere or compile them yourself.

If you are new to Tomcat, do not bother with these components to start with.
If you do use them, do not forget to read their documentation.

Apache Tomcat Native library

It is a library that allows to use the "Apr" variant of HTTP and AJP
protocol connectors in Apache Tomcat. It is built around OpenSSL and Apache
Portable Runtime (APR) libraries. Those are the same libraries as used by
Apache HTTPD Server project.

This feature was especially important in the old days when Java performance
was poor. It is less important nowadays, but it is still used and respected
by many. See Tomcat documentation for more details.

For further reading:

 - Apache Tomcat documentation

    * Documentation for APR/Native library in the Tomcat User's Guide

      https://tomcat.apache.org/tomcat-9.0-doc/apr.html

    * Documentation for the HTTP and AJP protocol connectors in the Tomcat
      Configuration Reference

      https://tomcat.apache.org/tomcat-9.0-doc/config/http.html

      https://tomcat.apache.org/tomcat-9.0-doc/config/ajp.html

 - Apache Tomcat Native project home

      https://tomcat.apache.org/native-doc/

 - Other projects

    * OpenSSL

      https://www.openssl.org/

    * Apache Portable Runtime

      https://apr.apache.org/

    * Apache HTTP Server

      https://httpd.apache.org/

To disable Apache Tomcat Native library:

 - To disable Apache Tomcat Native library when it is installed, or
 - To remove the warning that is logged during Tomcat startup when the
   library is not installed:

   Edit the "conf/server.xml" file and remove "AprLifecycleListener" from
   it.

The binary file of Apache Tomcat Native library is usually named

  - "tcnative-1.dll" on Windows
  - "libtcnative-1.so" on *nix systems

Apache Commons Daemon

Apache Commons Daemon project provides wrappers that can be used to
install Apache Tomcat as a service on Windows or as a daemon on *nix
systems.

The Windows-specific implementation of Apache Commons Daemon is called
"procrun". The *nix-specific one is called "jsvc".

For further reading:

 - Apache Commons Daemon project

      https://commons.apache.org/daemon/

 - Apache Tomcat documentation

    * Installing Apache Tomcat

      https://tomcat.apache.org/tomcat-9.0-doc/setup.html

    * Windows Service How-To

      https://tomcat.apache.org/tomcat-9.0-doc/windows-service-howto.html

The binary files of Apache Commons Daemon in Apache Tomcat distributions
for Windows are named:

  - "tomcat9.exe"
  - "tomcat9w.exe"

These files are renamed copies of "prunsrv.exe" and "prunmgr.exe" from
Apache Commons Daemon distribution. The file names have a meaning: they are
used as the service name to register the service in Windows, as well as the
key name to store distinct configuration for this installation of
"procrun". If you would like to install several instances of Tomcat 9.0
in parallel, you have to further rename those files, using the same naming
scheme.