

TOMCAT – MANUAL, AUTOMATIC, AND REMOTE START-UP

At first glance, starting up the **Tomcat** Server is a simple task. On the other hand, when you consider that it can be accomplished in a variety of ways (***manually, automatically, or remotely***), that these methods differ from platform to platform, and that when it comes down to it, if you can't get your server to start up, you're in a real bind, there's actually a lot to talk about.

The goal of this guide is to put all the information about starting **Tomcat** there is on one page, in a clear, accessible format.

To keep things simple, we've divided the guide into three large sections: **Manual Startup**, **Automatic Startup**, and **Remote Start-up**, with additional platform- and method-specific sub-sections as needed. As it's a closely related topic, the guide also includes a discussion about methods of reliably restarting your **Tomcat** server as an appendix.

MuleSoft's Tcat, an enterprise **Tomcat** solution, provides reliable ***remote startup and shutdown for single and multiple Tomcat servers***. Tcat is available as a free download from [MuleSoft's website](#), and can be evaluated for free all the way through pre-production.

Manual Start-Up

The simplest method of starting **Tomcat** is to manually start the server, either from the command line or by using a platform-specific method. Here's a run down of all the common methods of manually starting **Tomcat**.

Before attempting to start **Tomcat** by any of these methods, make sure that you are not already running another **Tomcat** instance with conflicting **HTTP**, **Server**, or **Connector** port configurations. Two instances cannot share the same port numbers.

Additionally, ensure that no other servers are already using ports 80 or 8080.

Both of these situations can cause **Tomcat** start-up to fail.

Using the Tomcat Shell Scripts

The shell scripts located in "**CATALINA_HOME/bin**" are the most bare-bones way of getting **Tomcat** up and running. The two scripts capable of starting **Tomcat** in this directory are named "**catalina**" and "**startup**", with extensions that vary by platform.

For **Unix-based systems**, these are standard shells scripts, with the extension ".sh". For **Windows** users, these scripts are included as batch files, with the extension ".bat".

"**Catalina**" is the script that is actually responsible for starting **Tomcat**; the "**startup**" script simply runs "**catalina**" with the argument "**start**" ("**catalina**" also can be used with the "**stop**" parameter to shut down **Tomcat**).

Most users will find it easiest to simply run "**startup**" from the command-line, which will start **Tomcat** normally, with output and error streams being written to the standard **Catalina.out** log file.

However, if you'd like to watch the server start up right in the terminal, you can use "**catalina**" with the "**run**" parameter to prevent these log streams from being redirected.

A number of other parameters can be used with "**catalina**" as well; notably, these include "jpda start", used to start Tomcat as a Java Platform Debugger Architecture for remote debugging, and "**-config [path/to/alt/server.xml]**", which allows you to specify an alternate "server.xml" configuration file to use during start-up.

A complete list of "catalina" parameters is available on the Apache project [website](#).

Before we move on, a quick tuning tip - since "**catalina**" calls the java command used to start Tomcat, you can set **JVM** options such as heap memory size in the **JAVA_OPTS** environment variable to have **Tomcat** automatically pass settings to the **JVM** when it starts. For more tuning tips, including a few tricks you can use to help Tomcat start up more efficiently, check out our **Tomcat** Performance Tuning guide.

Manual Start-Up For Linux-specific Distributions

If you chose to use the **RPM Package Manager** to install **Apache Tomcat** on your **Linux** machine, an init script was also installed to your init.d directory to handle Tomcat's startup and shutdown.

If you installed the standard RPM distribution, start **Tomcat** with the following command:

```
# /etc/rc.d/init.d/tomcat start
```

For users of the **Tomcat** distribution provided by JPackage.org, the script's name is appended by the [version number]:

```
# /etc/rc.d/init.d/tomcat[#] start
```

Manual Start-Up For Solaris-specific Distributions

If you installed Tomcat on your Solaris machine using the Blastwave distribution, an init script was also added to your init.d directory to handle Tomcat's startup and shutdown. You can start **Tomcat** with the following command, appending the appropriate [version number]:

```
# /etc/init.d/cswtomcat[#] start
```

Manual Start-Up For FreeBSD Distributions

The **FreeBSD** port of **Tomcat** requires a little more configuration than other system-specific distributions. Before you can use the **RCng** script that is packaged with the distribution to start **Tomcat**, you must first enable the script in **/etc/rc.conf** with the following additional line:

```
tomcat60_enable="YES"
```

After doing this, you can run **tomcat[version number].sh** with the following command:

```
# /path/to/etc/rc.d/tomcat[#].sh
```

Note that by default, this script is installed in `usr/local/`.

Manual Start-Up For Windows Distributions

There are two ways of manually starting Tomcat on a Windows machine; one is to execute "`catalina.bat start`" or "`startup.bat`" from the command line.

The other way, if you've installed **Tomcat** as a service, is to use the manager tool ***Tomcat6w.exe***, a small **GUI** program that resides in the task bar after it has been run. Double-clicking its icon will open a tabbed dialog window, which includes a start-up button, and can also be used to configure a few start up options.

If you don't know if you've installed **Tomcat** as a **Windows Service**, don't worry - this is covered in the **Automatic Start-Up** section. However, as a quick rule of thumb, if you installed **Tomcat** using the **Windows EXE** installer, **Tomcat** was also installed as a service automatically.

Manual Start-Up For Mac OS X Distributions

Technically, there is no "**Mac-specific**" distribution of **Tomcat** - Mac users simply download the ***binary distribution*** and unpack it. Thus, all you need to do to start **Tomcat** on a Mac is to run "`catalina.sh start`" or "`startup.sh`" within **Terminal**.

However, the **OS X** user structure allows you to start **Tomcat** in a more secure manner by adding some additional parameters to the command:

```
$ cd /; sudo -u nobody /path/to/tomcat/bin/startup.sh; cd -
```

This command allows you to run **Tomcat** as the unprivileged system user "***nobody***", which will deny **Tomcat** access to folders and directories it doesn't need. The additional `cd` commands aim to prevent access errors that can be caused by the combination of "***nobody***"'s limited access and the shell script's directory traversal methods by calling the script from the root folder.

Start Automatically on Boot

An important step of transitioning from a **Tomcat** development environment to a production setting is configuring **Tomcat** to automatically start on system boot. This ensures that if a machine must restart for any reason (***applying an update, etc.***), the server will come back online, even if it is being used in a blade environment without a screen or user to interact with it, and also offers some additional security.

Starting Tomcat Automatically on a FreeBSD Machine

We'll start with **FreeBSD** machines, because there's actually no additional steps required. If you've enabled "`tomcat[#].sh`" in "`/etc/rc.conf`", as outlined in the previous section, FreeBSD will start Tomcat automatically when you boot by default.

Starting Tomcat Automatically on a Solaris Machine

Getting **Tomcat** starting on boot for **Solaris** machines is just as easy as on **FreeBSD** machines - that is, if you installed the **Blastwave** distribution, which is configured for automatic startup by default.

If you installed the binary distribution provided by **Apache**, you'll have to do a little more work. First, follow the instructions provided below for **Linux** users to create a custom init script, which will be used to automatically start **Tomcat**.

Once you've done this, save your script in a new directory of `"/etc/init.d"` (we recommend `"tomcat"`), and configure the proper permissions settings (***the process must belong to the sys group and be owned by root, and both sys and root must be able to read and execute the script***).

Now, simply link the service to **rc3.d** (you'll need to substitute a start order [value] appropriate for your system):

```
#ln -s /etc/init.d/tomcat /etc/rc3.d/S[#]tomcat
```

Starting Tomcat Automatically on a Windows Machine

Automatic startup on **Windows** machines is accomplished by running **Tomcat** as a **Windows Service**, a background process that gathers all processes associated with a given application together in one container to be managed by the system. **Windows** services can be configured to boot automatically at login, and shut down smoothly on logout.

As mentioned in the manual start-up section for **Windows** distributions, if you utilized the **Tomcat EXE** installer package to install Tomcat on your **Windows** machine, you have already installed Tomcat as a service. You can configure the server to start automatically either through the tomcat6w.exe tray program, or by editing the preferences for "Apache Tomcat" in the Services configuration pane, located in the **Administrative Tools** section of the **Windows Control Panel**.

You can set the Tomcat service startup option to either **"Automatic"**, if you want the service to start immediately, or **"Automatic (Delayed)"**, which aims for a faster overall system boot by delaying the startup of non-system-critical services until all core components have been loaded.

If you did not use the **Windows** installer, you can install **Tomcat** as a service manually.

Starting Tomcat Automatically on a Linux Machine

The correct way to configure automatic start-up in a Linux environment depends on how you installed **Tomcat**. As mentioned above, both RPM distributions of **Tomcat** come packaged with custom init scripts, which the system can be configured to run on login. The standard binaries do not, so you'll have to write your own init script. If you don't know how to do this, there is a good Tomcat-specific guide available [here](#).

Once the proper init script is in place, the same process is used to start **Tomcat** on boot for all distributions. **Linux** uses **"run levels"** to determine the order in which various services should be started. Anything above a 1 or below a 6 (***the shutdown and boot level***) is a multi-user level, meaning that it can handle the kind of connections **Tomcat** needs to run. To make **Tomcat** start automatically when the system is in levels 2,3,4, or 5, use the **"chkconfig"** tool from the command line:

```
# chkconfig --level 2345 [name of the tomcat init script] on
```

Substitute the name of your init script (*either custom or as noted in the previous section*) for the placeholder.

Start Tomcat Automatically on a Mac OS X Machine

As of OS X version 10.4, **Apple** launches and manages all its init scripts via a program "**launchd**", which uses property list files to provide centralized access and configuration control over all the daemons available for the system to run.

This feature is designed for ease of use, but unfortunately, it causes Tomcat's start-up scripts some problems, as they are designed to run once and quit, without having to interact with any central managing utility. To prevent forking, you'll have to create a custom **.plist** file for Tomcat containing an init script workaround.

Fortunately, you won't have to do all this work yourself. The **Atlassian** product Confluence runs on Tomcat, and their documentation includes information about starting and stopping **Tomcat** automatically on **Mac OS X**, including script and **.plist** examples - you can find them [here](#). Simply remove all **Confluence-specific** references to get to get these scripts working for you!

A Word On Remote Start-Up

As **Tomcat**'s startup is controlled with simple shell scripts for most systems, it is possible to start **Tomcat** remotely if you have **SSH** access to the host machine. Here's an example of how you might do this on a **Unix-based machine**:

```
#ssh user@host.com
login: user
password: password
#cd $CATALINA_HOME/bin;
#sudo ./catalina.sh start;
```

You could also use this technique to restart Tomcat, by calling the startup and shutdown scripts in a single command:

```
#sudo ./catalina.sh stop;sudo ./catalina.sh start;
```

There are a few downsides to using this technique. For one thing, you'll have to configure SSH keys for every machine you want to start, which can be a huge hassle, even if you use a utility. Secondly, it's quite unreliable, as it won't provide you with any way of monitoring the startup.

If you're doing a restart, you'll be completely in the dark while the server is down. Lastly, starting and stopping **Tomcat** in this way offers no provision for lost threads or corrupted user data, meaning that problems may not appear until they are causing errors for your users.