MuleSoft™

Tcat Server User's Guide

Version 6 R2
December 2009

Confidential

The ideas contained in this publication are subject to use and disclosure restrictions as set forth in the license agreement.

Copyright

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of MuleSoft, Inc. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the agreement.

In addition, MuleSoft, Inc makes no representation or warranties either express or implied, with respect to this manual and accompanying software and specifically disclaim any implied warranties of merchantability or fitness for any particular purpose. This manual and accompanying software are sold “as is” and MuleSoft, Inc will in no event be liable for direct, indirect, incidental or consequential damages resulting from any defect, error or failure to perform except as expressly set forth in the license agreement.

Trademarks

Part number: 6.2en_us2009.12.7

Table of Contents

Preface

The *Tcat Server User's Guide* introduces Tcat Server from MuleSoft. It provides the conceptual information and instructions that you need to get started using Tcat Server.

Who Should Read This Guide?

This guide is intended for the following audiences:

- Decision makers who need to evaluate and understand Tcat Server
- Developers who create and monitor their Tomcat applications and servers

Typographic Conventions

The following table describes the typographic conventions used in the Tcat Server documentation:

Typeface	Meaning	Example
`AaBbCc123`	Files and directory names, parameters, command lines, and code examples.	Edit the information in `server.xml`
AaBbCc123	Placeholder text that you change.	`http://`*`serverName`*`:8080/console`
AaBbCc123	A live link to a web site, email address, or another section in the document	See page 6
AaBbCc123	The names of user interface controls, menus, and menu items.	Choose **File > Edit**.

Technical Support

If you have a paid subscription to MuleSoft, you can view the knowledge base and get assistance with Tomcat and Tcat Server at http://support.mulesoft.com. For information on purchasing a subscription, contact MuleSoft by phone at 1-877-MULE-OSS or by email at info@mulesoft.com.

Chapter 1

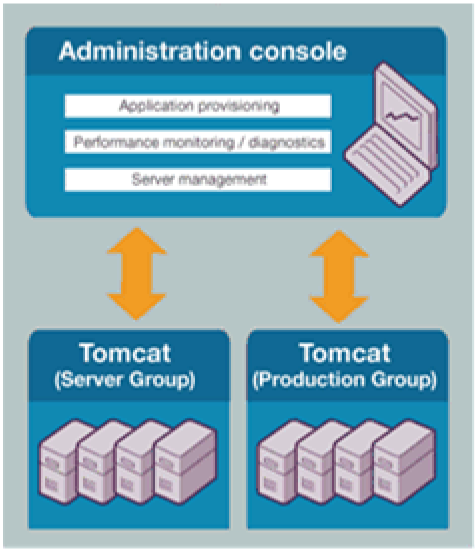
Introduction to Tcat Server

This chapter introduces Tcat Server and its underlying technologies. It includes the following sections:

- "What is Tcat Server?" on page 8
- "Understanding Web Applications and Containers" on page 9

What is Tcat Server?

Tcat Server from MuleSoft provides a powerful enterprise solution for developing, deploying, managing, integrating, and troubleshooting your web applications in Apache Tomcat. It comprises one or more Apache Tomcat instances, an administration console, and integration functionality (Mule iBeans).



Tcat Server provides several additional tools to make Tomcat even more powerful and simple to use. With Tcat Server, you can do the following:

- Use the Tcat Server Administration Console web application to monitor and control all your Tomcat instances on multiple servers. Unlike Tomcat's native manager application, the Tcat Server console allows you to quickly set up users and roles across all servers without having to edit the XML file in each instance separately.
- Easily deploy and view both remote and local applications from the console instead of just local instances.
- Monitor the health of the applications and the servers where you've deployed them.
- Troubleshoot problems.

Tcat Server is free to try out and use in development, so you can do a full proof of concept and test deployments before you go live. When you are ready to go into production, you purchase a subscription from MuleSoft at a fraction of what you would pay for a heavyweight application server.

Understanding Web Applications and Containers

This section provides a primer on the basics of web applications, servlets, and servlet containers.

About Web Applications

A *web application* (also called a *webapp*) is any application that can be accessed via an intranet or the Internet and displayed in a web browser. For example, you might have an application that allows customers to place orders from their web browsers on their home computers. When they place the order, the information is sent over the Internet to the web server, where the application is hosted. The key advantage of a webapp is that instead of being installed on each user's computer, it is installed on a web server, and multiple users (clients) can access it through their web browsers.

About Servlets

One type of webapp is a *servlet*. A servlet is written in Java and provides a specific set of functionality. For example, to provide order processing functionality, you might have a servlet that processes the customer's order, another servlet that routes the order to the shipping department, and so forth.

About Servlet Containers

A servlet is run and managed by a *servlet container* (also called *web containers*). The container provides a specific URL for the servlet and handles authentication between the requester and the servlet. When a request comes in, the container initializes the servlet, the servlet processes the request, and then the container stops the servlet until another request comes in.

About Tomcat

Tomcat is a very popular servlet container developed by Apache Software Foundation. Unlike the Apache Web Server, which is a C implementation of an HTTP web server, Tomcat is written in Java. Tomcat is free and is used in mission-critical environments around the world. If you are writing web applications in Java, Tomcat is a great choice for deploying and managing them.

Following are a few resources available to help you learn about Tomcat:

- Tomcat 6 Documentation: http://tomcat.apache.org/tomcat-6.0-doc/index.html
- Apache's Getting Started page for Tomcat: http://wiki.apache.org/tomcat/GettingStarted
- *Tomcat: The Definitive Guide* published by O'Reilly: http://oreilly.com/catalog/9780596101060/

The next chapter helps you get started using Tcat Server.

Chapter 2

Getting Started

This chapter describes how to get up and running with Tcat Server. It includes the following sections:

- "Quick Start" on page 11
- "Installing Tcat Server" on page 21
- "Starting and Stopping Tcat Server" on page 24
- "Modifying JAVA_OPTS" on page 27
- "Uninstalling Tcat Server" on page 27

Quick Start

This section describes a typical use cases for setting up and using Tcat Server to manage your Tomcat instances in less then ten minutes. It covers the following tasks:

- "Installing and Running Tcat Server" on page 11
- "Registering and Grouping Servers" on page 14
- "Deploying Applications" on page 15
- "Monitoring the Servers" on page 18

Installing and Running Tcat Server

Time to complete: three minutes (not including download)

You install the Tcat Server console and agent to the Tomcat instance where you want to manage your deployments. You then install the agent to all other Tomcat instances that you want to manage. These instructions assume you are installing the console and agent on your localhost and are installing a second agent on a remote server.

Prerequisites

Before you begin, do the following:

- Download the Tcat Server installer from http://www.mulesoft.com/download-tcat-server-enterprise-tomcat. It is available for Windows (`.exe`), Linux (`.sh`), and MacOS (`.dmg`). The files are less than 100 MB in size, so the download time will vary depending on your connection speed and which file you choose.
- Ensure that you have JDK/JRE 1.6 installed and that your `JAVA_HOME` or `JRE_HOME` environment variable is set.

Step One: Install Your Tcat Servers

In this step, you install one Tcat Server with the administrative console and one without.

To install the first Tcat Server:

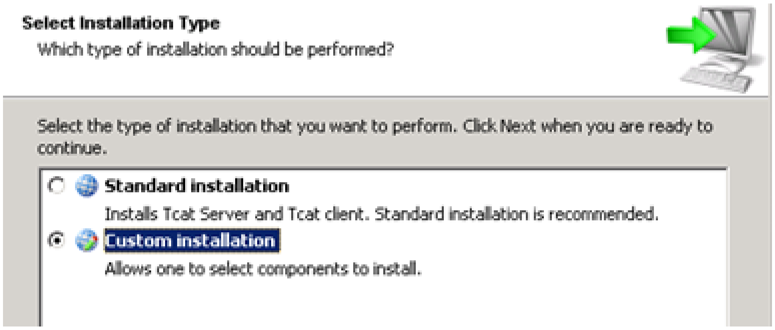
1. Run the installer on your local computer.
2. Select the standard installation option and accept all the defaults.
3. Run Tcat Server from the command prompt by navigating to the `bin` directory under your Tomcat home directory and then enter `tcat6 start`. (On Mac OS X, enter `startup.sh` instead.)

You have now installed the first Tcat Server with the administrative console and have started running the server. Do not close the command window, or Tcat Server will stop.

In the next step, you install the second Tcat Server, this time without the console.

To install the second Tcat Server:

1 Run the installer again and select the custom installation option, and then click **Next**.



Select Installation Type

Which type of installation should be performed?

Select the type of installation that you want to perform. Click Next when you are ready to continue.

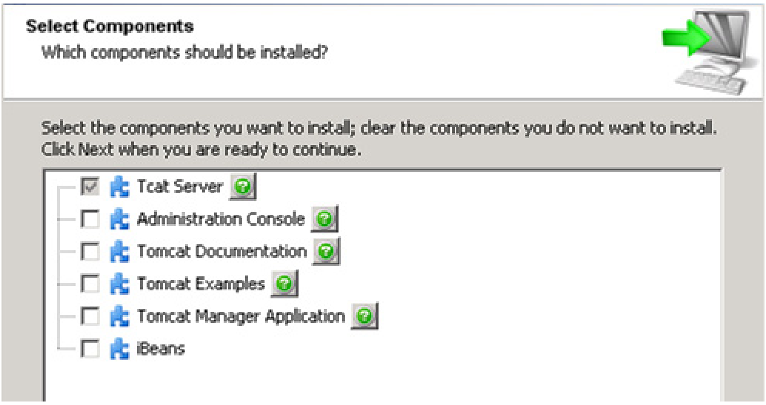
○ **Standard installation**

Installs Tcat Server and Tcat client. Standard installation is recommended.

● **Custom installation**

Allows one to select components to install.

2 Deselect the Administration Console option and click **Next**.



Select Components

Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☑ Tcat Server
- □ Administration Console
- □ Tomcat Documentation
- □ Tomcat Examples
- □ Tomcat Manager Application
- □ iBeans

3 Specify the location on the remote server where you want to install this Tcat Server instance, and then click **Next** and follow the remaining prompts.

4 Run this remote Tcat Server instance as described in step 3 on page page 12.

You now have two command windows open with Tcat Server running. Do not close these windows, or your Tcat Server instances will stop.

Step Two: Run the Tcat Server Console

1. Run the Tcat Server administration console by typing `http://localhost:8080/console` in your browser. If you specified a different port, use that instead of 8080.

2. Log in using `admin` for both the user name and password.

   Tcat Server provides a default password to make it easy to get started. You should change the password before going into production. For instructions, see the *Tcat Server Administrator's Guide*.

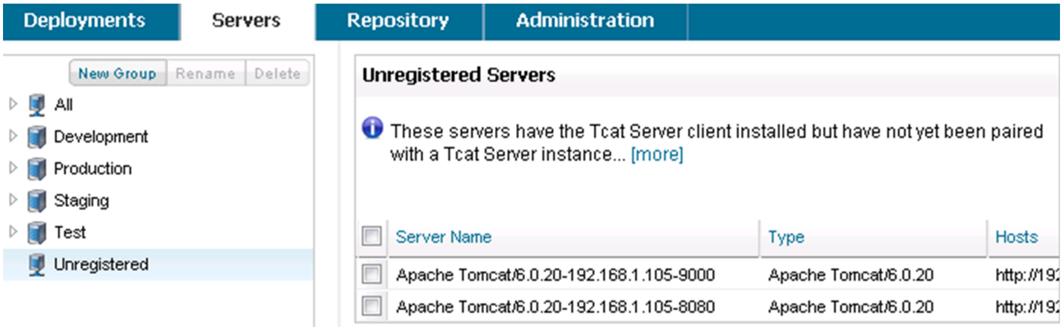
Registering and Grouping Servers

Time to complete: one minute

Each Tomcat instance that has the Tcat Server agent installed on it is referred to as a *server*. Before you can manage a server, you must register it in Tcat Server.

1. In the Tcat Server console, click the Servers tab, and then click **Unregistered** in the navigation tree on the left to display all servers that have not yet been registered with Tcat Server.

   Note If your servers do not appear in the Unregistered list, you can register them manually instead.



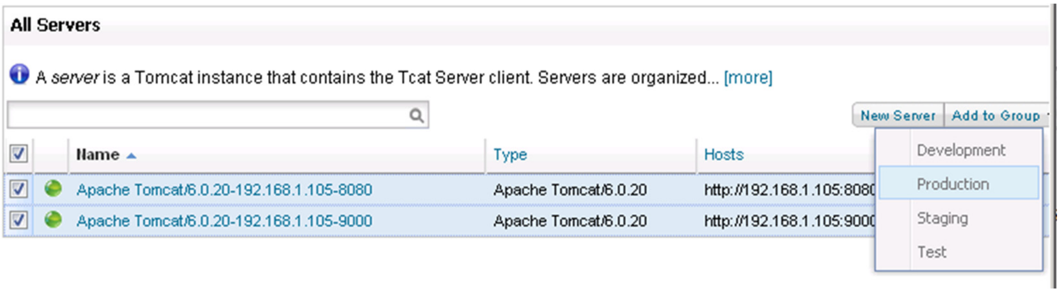
Unregistered Servers

These servers have the Tcat Server client installed but have not yet been paired with a Tcat Server instance... [more]

□	Server Name	Type	Hosts
□	Apache Tomcat/6.0.20-192.168.1.105-9000	Apache Tomcat/6.0.20	http://19
□	Apache Tomcat/6.0.20-192.168.1.105-8080	Apache Tomcat/6.0.20	http://19

2. Select the check box for both servers, and then click **Register**. The servers now appear when you click **All** in the navigation tree.

If you want to group some of your servers together, such as if you want to deploy the same application to all the servers in a particular group, you can create a server group (see "Creating a Server Group" on page 30) and add the servers you want to include in that group. For this example, click the servers' check boxes, click **Add to Group**, and then click the Production group.



All Servers

A *server* is a Tomcat instance that contains the Tcat Server client. Servers are organized... [more]

☑		Name ▲	Type	Hosts
☑	●	Apache Tomcat/6.0.20-192.168.1.105-8080	Apache Tomcat/6.0.20	http://192.168.1.105:8080
☑	●	Apache Tomcat/6.0.20-192.168.1.105-9000	Apache Tomcat/6.0.20	http://192.168.1.105:9000

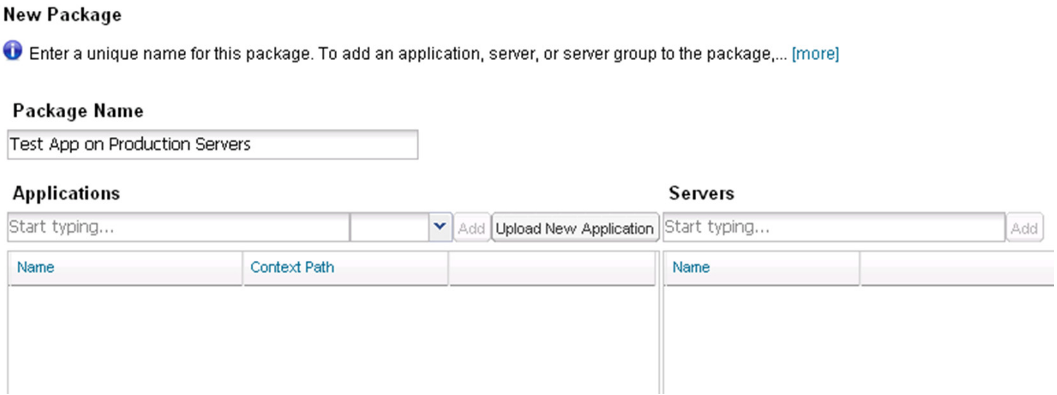
Deploying Applications

Time to complete: two minutes

After you have registered servers, you are ready to start deploying applications to them. To deploy an application, you create a package that specifies the application(s) to deploy and the servers or group(s) where you want to deploy them.

Step One: Start Creating the Package

1. Click the Deployments tab, and then click **New Package**.
2. Enter a unique name for this package.



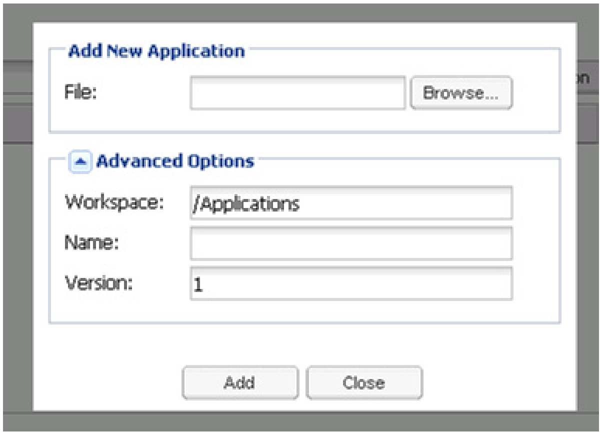
New Package

Enter a unique name for this package. To add an application, server, or server group to the package,... [more]

Package Name: Test App on Production Servers

Step Two: Add an Application to the Repository

1. Because you do not have any applications in your Tcat Server repository yet, click **Upload New Application** to add the first application you want to include in the package.
2. Browse to the location of any application (WAR) file you want to use for this example. You can also click the Advanced down arrow to specify a location, unique name, and version for the application as you want it to appear in the repository.



Add New Application

File:

Advanced Options

Workspace: /Applications

Name:

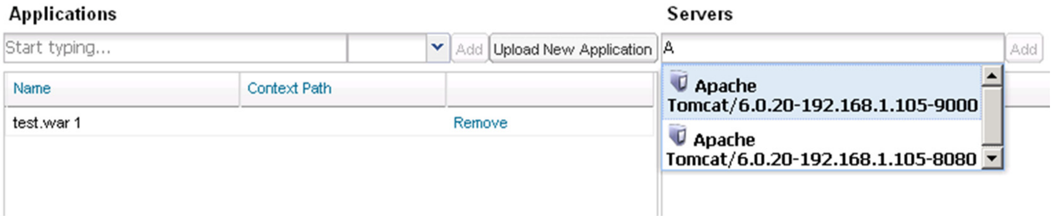
Version: 1

3 Click **Add** to upload the application to the repository and add it to the package.

In the future, you can add this application to a package simply by typing its name in the Applications list when creating a new package.

Step Three: Add Servers to the Package

Adding a server to a package is simple. Just type the name of the server in the Servers list, select it when it appears, and then click **Add**.



Name	Context Path	
test.war 1		Remove

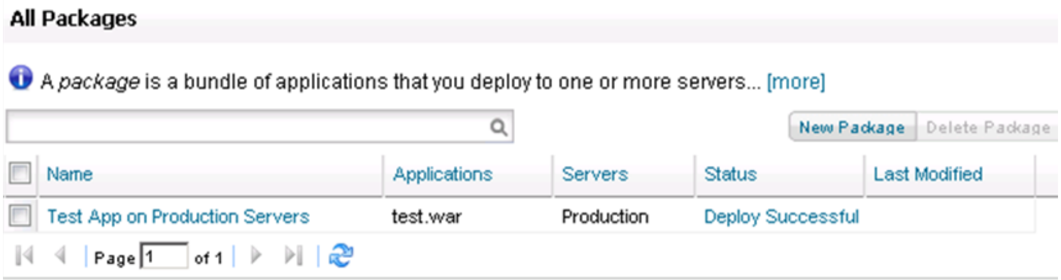
Servers: A

- Apache Tomcat/6.0.20-192.168.1.105-9000
- Apache Tomcat/6.0.20-192.168.1.105-8080

Alternatively, if you want to add a group to the package, type the name of the group instead. Adding a group to a package means that the applications in this package will be deployed to all the servers in that group. For this example, select the Production group to add all its servers to the package.

Step Four: Save and Deploy the Package

When you have finished adding applications and servers, click **Save and Deploy**. The application(s) in the package are copied to the webapp directories of the servers specified in the package, where their Tomcat instances immediately detect and deploy them.



A *package* is a bundle of applications that you deploy to one or more servers... [more]

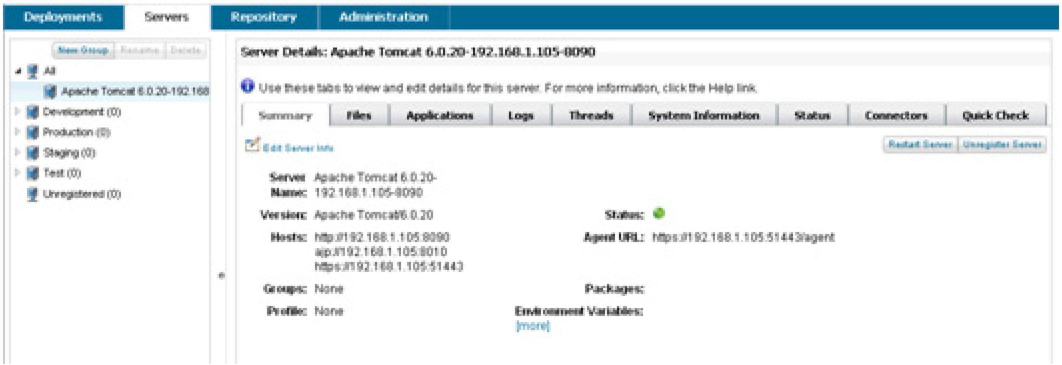
□	Name	Applications	Servers	Status	Last Modified
□	Test App on Production Servers	test.war	Production	Deploy Successful	

If you want to make changes to a package, simply click its name to view its details.

Monitoring the Servers

Time to complete: one minute

To monitor metrics such as memory usage for any of the registered servers, simply click the server's name on the Servers tab. The Server Details screen is divided into several tabs that display information such as logs, threads, up/down status, and details on the applications deployed to that server. For example, to drill down on a specific application, click its name on the Applications sub-tab. To perform a quick check on the server's health, click the Quick Check tab.



Server Details: Apache Tomcat 6.0.20-192.168.1.105-8090

Use these tabs to view and edit details for this server. For more information, click the Help link.

Server Name: Apache Tomcat 6.0.20-192.168.1.105-8090
Version: Apache Tomcat/6.0.20
Status: ●
Hosts: http://192.168.1.105:8090
ajp://192.168.1.105:8010
https://192.168.1.105:51443
Agent URL: https://192.168.1.105:51443/agent
Groups: None
Packages:
Profile: None
Environment Variables:

That's all there is to it! You are now managing your Tomcat instances with Tcat Server, allowing you to easily deploy applications to multiple instances at once, undeploy applications, check memory usage of individual servers, and more. The rest of the topics in this guide describe how to perform these tasks.

The next section describes installation of Tcat Server in more detail, including how to do manual installation.

Upgrading to Tcat Server 6 R2

This section provides important information for users who are upgrading from a previous release of Tcat Server 6 to the R2 release. If you are not upgrading, skip ahead to the next section.

Throughout this page, the directory of your Tomcat installation is referred to as CATALINA_HOME. For information on what has changed in this release, see the Release Notes at http://www.mulesoft.org/display/TCAT/Release+Notes.

Upgrading via the Installer

The R2 installer upgrades your Tcat installation automatically. It installs a new console and a new agent WAR inside the installation directory. It backs up the previous console and agent into the `CATALINA_HOME/tcat-backup-DATE` in case you need to roll back to your previous installation.

Typically, the installer requires no extra steps. However, if you are using LDAP or the EC2 plugin, you must perform some manual steps as described at the end of this section.

Manually Upgrading

To manually upgrade using the R2 installation ZIP file, you take the following steps.

1. Stop your Tomcat or Tcat Server instance.
2. Copy your existing Tomcat installation directory to a backup file. For example, on UNIX, type:

   ```
   $ cp -a CATALINA_HOME CATALINA_HOME.bak
   ```

3. Extract the R2 installation ZIP file. For example, on UNIX, type:

   ```
   $ unzip TcatServer-6.2.0.zip
   ```

4. Copy the new agent, console, and scripts to your Tomcat installation directory. For example:

   ```
   $ cd TcatServer-6.2.0
   $ cp webapps/* CATALINA_HOME/webapps/
   $ rm -rf CATALINA_HOME/webapps/console CATALINA_HOME/webapps/agent
   $ mkdir -p CATALINA_HOME/webapps/console CATALINA_HOME/webapps/agent
   $ unzip webapps/console.war -d CATALINA_HOME/webapps/console
   $ unzip webapps/agent.war -d CATALINA_HOME/webapps/agent
   $ cp CATALINA_HOME.bak/webapps/agent/WEB-INF/agent.properties
   CATALINA_HOME/webapps/agent/WEB-INF/
   $ cp CATALINA_HOME.bak/webapps/agent/WEB-INF/truststore.jks
   CATALINA_HOME/webapps/agent/WEB-INF/
   $ mkdir -p CATALINA_HOME/conf/Catalina/localhost
   $ cp conf/Catalina/localhost/tcat* CATALINA_HOME/conf/Catalina/localhost
   $ cp bin/tcat* CATALINA_HOME/bin
   $ cp bin/catalina.sh CATALINA_HOME/bin
   ```

5 If your original installation created a directory named `galaxy-data` anywhere other than in the root of your `CATALINA_HOME` directory, move it to the root of your `CATALINA_HOME` directory now. For example:

```
$ mv CATALINA_HOME/bin/galaxy-data CATALINA_HOME/galaxy-data
```

6 Clear out your `work` and `temp` directories. For example:

```
$ rm -rf CATALINA_HOME/temp/* CATALINA_HOME/work/*
```

7 Compare the following configuration files in your original installation with those in the new distribution: `server.xml`, `conf/catalina.properties`, `conf/web.xml`, `conf/context.xml`, `conf/tomcat-users.xml`, and `conf/logging.properties`. Where there are differences, you should consider whether you should merge those changes into your installation. For example, to compare `server.xml` on UNIX:

```
$ diff -u conf/server.xml CATALINA_HOME/server.xml
```

8 Set the proper ownership and group for the runtime tree (you must have superuser privileges to perform this step). For example:

```
chown -R tomcat:tomcat CATALINA_HOME
```

9 If you are using LDAP or the Amazon EC2 plugin, see the directions below on how to complete the upgrade.

You are now ready to start your Tcat Server.

LDAP Upgrade

To migrate your previous LDAP configuration file from your backup to the new Tcat Server installation, you must copy the LDAP configuration file to the new installation and then copy the LDAP JAR to the Console webapp.

Copy the LDAP Configuration File

If you used the installer to upgrade, you restore your configuration by copying `tcat-backup-DATE/console/WEB-INF/classes/ldap.xml` to the `CATALINA_HOME/webapps/console/WEB-INF/classes` directory. For example, on UNIX, type:

```
$ cd CATALINA_HOME
$ cp tcat-backup-DATE/console/WEB-INF/classes/ldap.xml
webapps/console/WEB-INF/classes
```

If you upgraded manually, you restore your configuration by copying `webapps/console/WEB-INF/classes/ldap.xml` from your backup file to the `CATALINA_HOME/webapps/console/WEB-INF/classes` directory. For example, on UNIX, type:

```
$ cd CATALINA_HOME
$ cp CATALINA_HOME.bak/webapps/console/WEB-INF/classes/ldap.xml
webapps/console/WEB-INF/classes
```

Copy the LDAP JAR into the Console Webapp

You must install the new LDAP JAR into the Console webapp as described in the *Tcat Server Administrator's Guide*.

Amazon EC2 Plugin Upgrade

The Amazon EC2 plugin for Tcat R1 is not forward compatible with R2. You must download a new version of the plugin and install it by following the Amazon EC2 instructions in Chapter 4, "Deploying to Amazon EC2."

Installing Tcat Server

This section provides details on installing, starting, stopping, and uninstalling Tcat Server. If you are upgrading from a previous release, see "Upgrading to Tcat Server 6 R2" on page 18 instead.

Prerequisites

Ensure that you have JDK or JRE 1.5 or later installed and that your `JAVA_HOME` or `JRE_HOME` environment variable is set correctly. For example, on Windows, choose the System utility from the Control Panel, click the Advanced tab, click **Environment Variables**, and then add the `JAVA_HOME` or `JRE_HOME` system variable.

Automated Installation

The Tcat Server installer allows you to install Tomcat, the Tcat Server agent, the Tcat Server administration console, and Mule iBeans on Windows, Linux, and Mac OS X computers. It installs a preconfigured version of Tomcat, saving you some manual steps.

‣ To install Tcat Server:

1. On the computer where you want to install your master Tcat Server instance, download the installer from http://www.mulesoft.com/download-tcat-server-enterprise-tomcat, and then run it and accept all the default settings.
2. On each computer where you want to run a Tomcat instance that you will manage through the administration console, run the installer again, this time deselecting the administration console. If you are installing multiple Tcat Servers on the same computer, be sure to specify unique ports for each as described in "Installing Multiple Tcat Server Instances on a Single Computer" on page 23.

Manual Installation

If you do not want to use the installer, such as if you are running Debian (which is not supported by the installer), follow the instructions in this section.

Installing Tcat Server

1. On the computer where you want to install your master Tcat Server instance and run the administration console, download and unpack the `ZIP` or `TAR.GZ` version of Tcat Server into any directory.

   Note If you have an existing instance of Galaxy running, do not deploy Tcat Server on that same computer.

2. On each computer where you want to run a Tomcat instance that you will manage through the administration console, copy the files from the first instance you created and delete the `console.war` file (and the `console` subdirectory, if present) from the `webapps` directory.

Alternatively, if you already have instances of Tomcat 6 installed that you want to manage, you can simply copy the `agent.war` file from the `webapps` directory of your Tcat Server installation to the `webapps` directory of each Tomcat 6 instance you want to manage. If Tomcat is handling multiple hosts, you likely have multiple `webapps` directories as well, so be sure to copy the WAR to each `webapps` directory.

Installing Multiple Tcat Server Instances on a Single Computer

If you are installing multiple Tomcat instances on the same computer, ensure you have unique values for the following ports in *`tomcatHome`*`/conf/server.xml` (you can ignore those that are commented out if you don't require them in your implementation):

- Shutdown port
- HTTP connector port
- Any connector executor ports
- HTTPS connector port
- AJP connector port

You must also set the securePort to a unique value (see "Modifying the Secure Port" on page 24).

Lastly, if you are connected to your network via a virtual private network (VPN), disconnect before running Tcat Server. After you have registered all your Tcat Server instances, you can connect to your VPN again.

Installing Tomcat Separately

If you want to install additional instances of Tomcat and the Tcat Server agent separately, follow these instructions.

1. Download Tomcat 6 from http://tomcat.apache.org/download-60.cgi. You can also use Tomcat 5.5 (see step 5 below for additional instructions).
2. Unzip the file. On Windows, you have to rename the directory so that it doesn't have any periods.
3. Copy the `agent.war` file from your Tcat Server `webapps` directory to your new Tomcat `webapps` directory.
4. Type `catalina run` and press **Enter**. Do not close this command window, or Tomcat will stop.

5 If you are running Tomcat 5.5, navigate to your Tomcat directory and copy the `/server/webapps/manager/WEB-INF/lib/catalina-manager.jar` file to the `/webapps/agent/WEB-INF/lib/` directory.

6 Ensure that your HTTP connector port is set to an unused port in `conf/server.xml`. The HTTP connector port is required for connecting your Tomcat instance to Tcat Server.

7 Launch your web browser and point to `http://localhost:8080`. If you see the welcome screen, you installed Tomcat correctly. If another app is using port 8080, you can change the port in the `/conf/server.xml` file under your Tomcat home directory.

8 Repeat these steps for each additional Tomcat server you want to run. If you want to install multiple Tomcat instances on the same computer, ensure you have unique values for the ports described in the previous section.

Modifying the Secure Port

The Tcat Server agent (also called the *client*) allows the web applications in a Tomcat instance to be discovered, deployed, and monitored via the Tcat Server administration console. When the agent is deployed to a Tomcat instance, it becomes a Tcat Server instance (or just *server*) and broadcasts the location of the web applications on a multicast channel. The console listens on that channel and sends its security certificate to the agent, which authenticates the certificate and establishes an SSL connection with the console. Each agent authenticates and pairs with only one console, whereas each console can have multiple agents paired with it. The agent sends the information about the web applications over SSL, creating a secure channel of communication between Tomcat and the console.

After the server and the console have created a secure channel, the server appears in the Unregistered group on the Servers tab in the console. The default port of 51443 is used for the secure channel on which the agent is discovered. If you need to specify a different port, modify the `securePort` value by adding `-Dtcat.securePort=51444` to your `JAVA_OPTS` environment variable (see the next section), replacing 51444 with the port you want to use.

Starting and Stopping Tcat Server

This section describes how to start and stop Tcat Server. It contains the following sections:

- "Starting and Stopping on Windows and Linux" on page 25
- "Additional Options on Windows" on page 25

- "Additional Options on Linux" on page 26
- "Starting and Stopping on Mac OS X" on page 26
- "Starting the Administration Console" on page 26

Note If you want to use the standard Tomcat catalina or startup commands to start Tomcat, any environment variables you set in the Tcat Server console will not take effect until you restart Tomcat using the Tcat Server commands. Therefore, for best results when setting environment variables in the Tcat Server console, you should always start Tomcat using the Tcat Server commands described in this section.

Starting and Stopping on Windows and Linux

To start Tcat Server, navigate to the `bin` directory and enter the following at the prompt:

```
tcat6 start
```

or prefix `tcat6` with the path to the `bin` directory to run the command from a different directory. To start the administration console, see page 26.

To stop Tcat Server, simply close the command window, or use:

```
tcat6 stop
```

You can also restart the server:

```
tcat6 restart
```

and get the server's status and process ID:

```
tcat6 status
```

Additional Options on Windows

If you installed Tcat Server via the installer, you can choose **Start Tcat Server** and **Stop Tcat Server** from the Tcat Server 6 group in the Windows Start menu.

To start the administration console, see page 26.

Additional Options on Linux

If you installed as a non-root user via the installer, you can use the graphical desktop applications menu to start, stop, or restart the server.

If you installed as root via the installer, you can use the init script:

```
service tcat6 start
```

If the service command isn't available, use the following command instead:

```
/etc/init.d/tcat6 start
```

If you installed using the ZIP file instead of the installer and you have root privileges, you can symlink the init script into `/etc/init.d` so that you can invoke it using standard init script conventions (such as using the `service` command). For example:

```
# ln -s /opt/TcatServer6/conf/Catalina/localhost/tcat6-linux.sh
/etc/init.d/tcat6
```

To start the administration console, see page 26.

Starting and Stopping on Mac OS X

Navigate to the Tomcat `bin` directory and enter the following command at the terminal prompt:

```
startup.sh
```

To stop a Tcat Server instance, enter the following command:

```
shutdown.sh
```

Starting the Administration Console

To run the administration console, enter `http://localhost:8080/console` in your web browser, replacing `localhost:8080` with the correct server name and port where the console is deployed. You can now select and register one or more of the unregistered servers, adding them to server groups as needed. For more details, see "Working with Servers" on page 28.

Modifying JAVA_OPTS

There are several reasons to modify your `JAVA_OPTS` environment variable:

- You want to enable JMX so that you can get more detailed information about connectors and server status, e.g., `-Dcom.sun.management.jmxremote`
- You need to increase your memory settings because you are installing all the components offered in the installer, e.g., `-Xmx512M -XX:PermSize=64M -XX:MaxPermSize=128M`
- You need to modify the secure port (see "Modifying the Secure Port" on page 24), e.g., `-Dtcat.securePort=51444`

After installing Tcat Server, you can modify `JAVA_OPTS` using the Tcat Server console, either by setting the options manually on each server by modifying the server's environment variables or (see "Working with Environment Variables" on page 39) or, if you have administrative privileges, by setting them in a server profile that you use across multiple Tcat Server instances (see the *Tcat Server Administrator's Guide*).

Uninstalling Tcat Server

To uninstall Tcat Server, you have the following options:

- If you installed Tcat Server on Windows via the installer, choose **Uninstall Tcat Server** from the Windows Start menu.
- If you manually installed Tcat Server and Tomcat in the same directory, and you want to delete both programs, simply delete the entire folder.
- If you manually installed Tcat Server on an existing Tomcat installation, delete the console, agent, and iBeans (if applicable) webapps and their folders from the `webapps` directory.

Chapter 3

Working with Servers and Applications

This chapter describes how to work with Tcat Server. It includes the following sections:

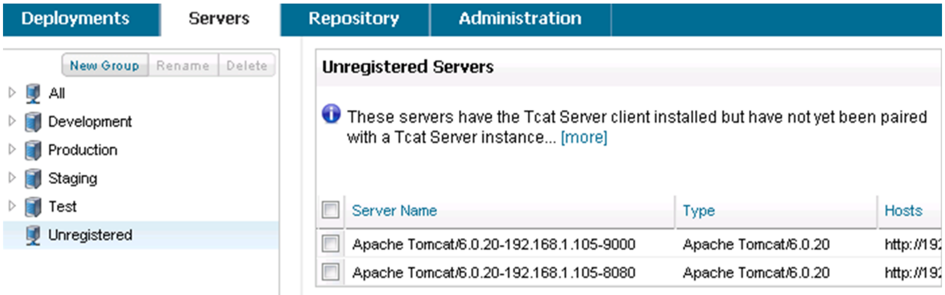
- "Working with Servers" on page 28
- "Deploying Applications" on page 34
- "Monitoring a Server" on page 38
- "Monitoring Applications" on page 49
- "Troubleshooting" on page 52

Working with Servers

The Servers tab displays the Tomcat instances that contain the Tcat Server client, allowing you to manage them with Tcat Server. Servers are organized into *groups*. Click a group in the navigation tree on the left to see its servers, or click **All** to see all servers. Click a server's check box to perform actions on it such as adding it to a group, or click the server's name to view its details.

Registering a Server

When you add the Tcat Server client to a Tomcat instance, that instance appears as a server in the Unregistered group on the Servers tab. You can then register this server so that you can manage it with Tcat server.



Unregistered Servers

These servers have the Tcat Server client installed but have not yet been paired with a Tcat Server instance... [more]

□	Server Name	Type	Hosts
□	Apache Tomcat/6.0.20-192.168.1.105-9000	Apache Tomcat/6.0.20	http://19
□	Apache Tomcat/6.0.20-192.168.1.105-8080	Apache Tomcat/6.0.20	http://19

Note If your network does not support multicasting or you are connected via VPN, your unregistered servers do not appear in the Unregistered group, and you must register the servers manually instead (see "Manually Adding a Server" on page 34).

To register a server from the Unregistered group:

1. Select the check box for one or more unregistered servers and click **Register**. If you want to add it to a server group as you register it, click **Register and Add to Group** instead and then select the group.
2. Optionally, click a server's name and type a new name for it. Each name must be unique.
3. When you have finished renaming servers, click **Save**.

The newly registered server now appear on the Servers tab when you click **All** in the navigation tree. To add them to a specific group, see the next section.

Using a Server Profile

A *server profile* is a set of environment variables and files that you can publish to multiple servers at once. You can use the profile to override variables for the operating system, user, and application on a Tomcat instance. You can set the server profile when you are viewing a server's details, or you can set the profile for multiple servers by selecting them on the Servers tab, clicking **Set Profile**, and then selecting the profile you want them to use. You can set only one profile per server.

If the server profile contains environment variables that affect the startup of Tomcat, you must restart the servers before these variables will take effect (see below).

Note If you change environment variables outside of Tcat Server and want them to take effect on your Tomcat instance, you must restart Tomcat manually outside of Tcat Server, as Tcat Server uses the environment variables from the current JVM plus those specified in the profile to restart Tomcat.

For information on creating a server profile, see the *Tcat Server Administrator's Guide*.

Restarting a Server

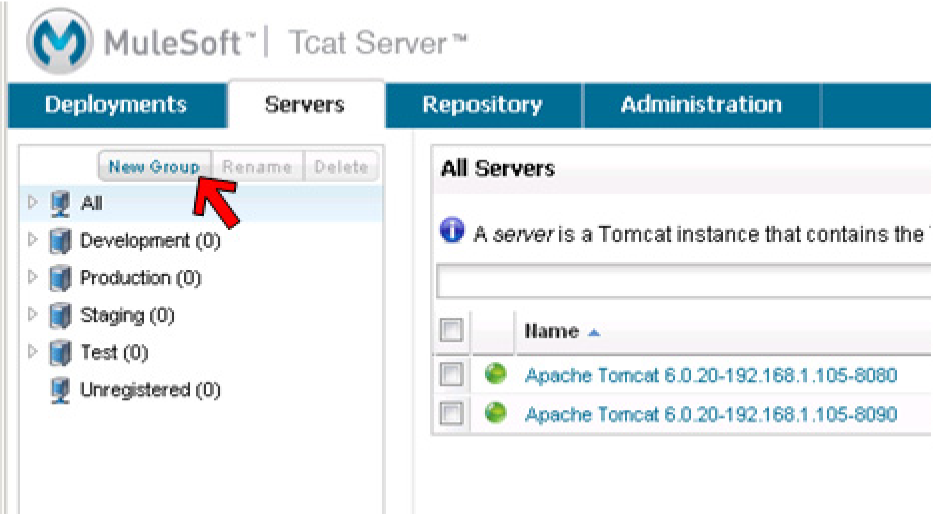
You can restart one or more servers by selecting their check boxes and clicking **Restart**. You can also restart a server when you are viewing its details.

Creating a Server Group

A *server group* is simply a logical grouping of servers. The All group lists all registered servers, and the Unregistered group lists any servers that contain the Tcat Server client but are not yet being managed by Tcat Server (see "Registering a Server" on page 29).

‣ To create a server group:

1 In the navigation pane on the left side of the Servers tab, click **New Group**.



All Servers

A *server* is a Tomcat instance that contains the

□		Name ▲
□	●	Apache Tomcat 6.0.20-192.168.1.105-8080
□	●	Apache Tomcat 6.0.20-192.168.1.105-8090

2 Enter a name for this group, and then click **OK**.

You can now start adding servers to this group. You can also deploy packages to the group (see "Deploying Applications" on page 34), allowing you to deploy all the package's applications to all the group's servers in one step.

Renaming a Group

To change the name of a group, select it in the navigation tree, click **Rename**, enter the new name, and click **OK**.

Finding a Server

There are several ways to find a server in the Tcat Server Administration Console:

- Type a server's name in the search box at the top of the Servers tab and then select it from the list that appears.
- If you know which group the server is in, click the group's name in the navigation tree on the left, and then browse through the list of servers in that group.

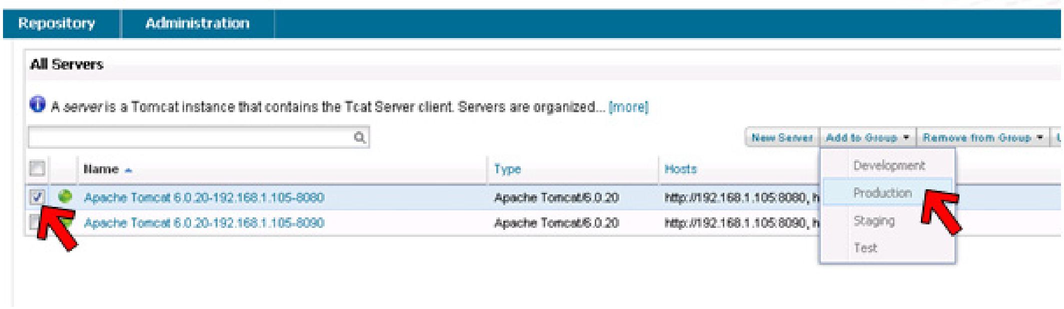
- If the server is not a member of a group, click **All** in the navigation tree to see all servers.
- If the server is not yet registered to be managed by Tcat Server, click **Unregistered**.

Adding Servers to a Group

Server groups provide a convenient way to organize your servers, such as those that are in development and those that are in production. You can deploy applications to a server group instead of having to deploy to each individual server in that group. When you deploy applications to a server group, each server you subsequently add to that group will immediately have those applications deployed on it.

To add a server to a group:

1. On the Servers tab, select the check box of one or more servers you want to add to a group.
2. Click **Add to Group**, and then select the group from the drop-down list.



All Servers

A server is a Tomcat instance that contains the Tcat Server client. Servers are organized... [more]

□		Name	Type	Hosts
☑	●	Apache Tomcat 6.0.20-192.168.1.105-8080	Apache Tomcat/6.0.20	http://192.168.1.105:8080, h
□	●	Apache Tomcat 6.0.20-192.168.1.105-8090	Apache Tomcat/6.0.20	http://192.168.1.105:8090, h

3. At the confirmation screen, click **Yes** to add the selected servers to the group and deploy that group's applications to the servers.

Any applications deployed to this group are now deployed to the servers you added to the group. To undo this action, you can remove one or more servers from the group.

Removing Servers from a Group

When you remove a server from a group, any applications deployed to the group are undeployed from that server, and the server no longer appears when you click that server group in the navigation tree.

To remove servers from the group:

1. Select the check box of one or more servers you want to remove from the group.
2. Click **Remove from Group**, and then select the group from the drop-down list.
3. At the confirmation screen, click **Yes** to remove the selected servers from the group and undeploy that group's applications from the servers.

Deleting a Server Group

If you no longer need a server group, you can delete it from the system. Deleting a server group does not delete the servers in that group but simply deletes the grouping. However, it does undeploy all the applications that were deployed to the group.

To delete a server group, select that group in the navigation tree, click **Delete**, and then click **OK** to confirm that you want to delete the group and undeploy all its applications from the servers in that group.

Unregistering a Server

If you no longer need a server in your server inventory, you can remove it from the repository.

To remove a server:

1. Select the check box of one or more servers you want to unregister.
2. Click **Unregister**.
3. At the confirmation screen, click **Yes** to unregister the selected server(s) so that you can no longer manage them with the Tcat Server Administration Console.

These servers no longer appear in the groups to which they were previously assigned nor in the All group.

If the console becomes unavailable and you want to unregister its servers so that you can register them with another console, you must manually delete the `webapps\agent\WEB-INF\truststore.jks` file under each server to unregister it. The server will then become available again for registering in another console.

Manually Adding a Server

In some cases, an unregistered server will not appear in the Unregistered group. This can happen if multicasting is not supported in your network, if you are connected via VPN, or if there was an error after you unregistered a server. In this case, you can register the server manually by clicking **New Server**. You then enter a name for the server and the server agent's URL, such as `http://localhost:8080/agent` (replacing `localhost` and `8080` with the correct host and port).

Deploying Applications

The key purpose of Tcat Server is to allow you to deploy one or more applications to multiple Tomcat instances, local or remote, in a single step, and to undeploy them just as easily. This page describes the various tasks related to deployment.

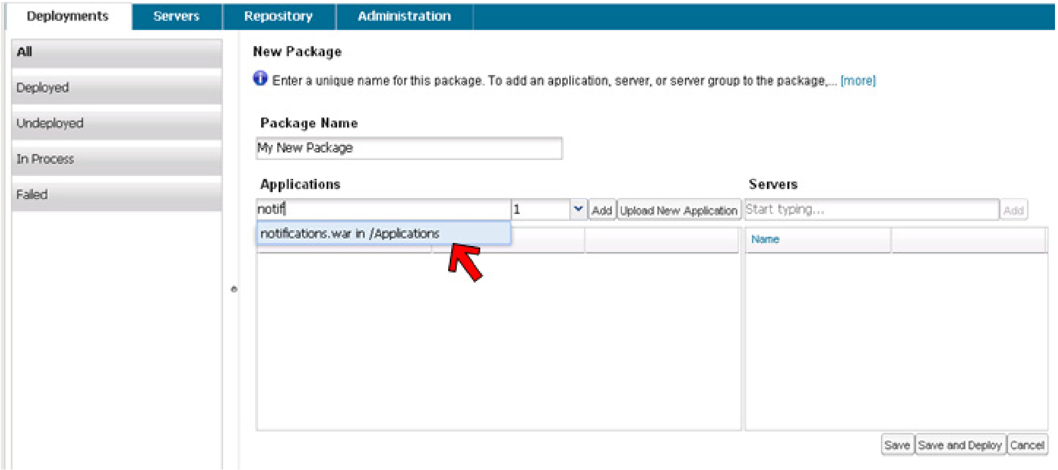
Creating a Package

A package is a group of applications that you deploy to one or more servers. When you create a package, you specify the applications to include in that package, and you select the target servers where you will deploy the package.

‣ To create the package:

1. On the Deployments tab, click **New Package**.
2. Enter a unique name for this package.

3 Under Applications, start typing the name of the first application you want to add, and click its name when you see it listed. If the application is not listed, you must upload it manually (see page 36) and skip the next step.



Enter a unique name for this package. To add an application, server, or server group to the package,... [more]

Package Name: My New Package

Applications: notif | 1

notifications.war in /Applications

Servers: Start typing...

4 In the version box, select the version of the application you want to deploy. If it has not been versioned yet, type 1. Click **Add.**

5 Repeat these steps for additional applications you want to include in the package. If you make a mistake, click **Remove** next to the application you want to remove from the package.

6 Under Servers, start typing the name of the server or group where you want to deploy the package, and click its name when you see it listed.

7 Click **Add**, and then repeat these steps for additional servers or groups where you want to deploy the package.

8 When you have added all the applications, servers, and groups you want, click **Save** if you want to save this package now but deploy later, or click **Save and Deploy** to deploy it now.

The package now appears on the Deployments tab. If you saved the package without deploying it, you can deploy it later by selecting its check box and clicking **Deploy**.

Uploading Applications

There are three ways to upload applications (WAR files) into the repository:

- Add an application manually when you are creating a package (see page 36).
- If you have administrator permissions, add applications directly into the repository using the Repository tab.
- When developing an application, upload it to the repository as part of the build process using the Maven plug-in (see the *Tcat Server Administrator's Guide*).

To upload an application manually:

1. When creating or editing a package on the Deployments tab, click **Upload New Application**.
2. Click **Browse** and navigate to the location of the application's WAR file and click **Open** to select it.
3. Optionally, click the Advanced down arrow to specify a location, unique name, and version for the application as you want it to appear in the repository.
4. Click **Add** to add the application to the repository and include it in the package.

For information on managing an application, such as moving it from one location to another in the repository, see the *Tcat Server Administrator's Guide.*

Modifying a Package

After you have created a package, you might want to modify some of its details, including its applications, configuration files, or target servers.

To modify a package:

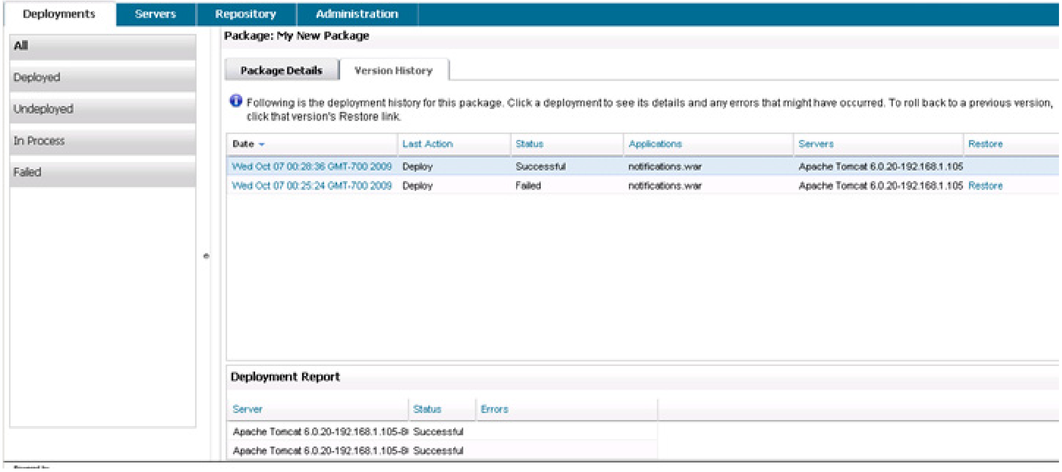
1. On the Deployments tab, click the name of the package you want to modify.
2. Make the changes you want to the package, and then click **Save** to save the changes without deploying the updated version yet (you can click **Deploy** later), or click **Save & Deploy** to save and deploy in one step.

Viewing a Package’s History

Each time you modify a package, deploy it, or undeploy it, a new version of the package is created in the package history. To see what has changed over time, you can view the package's history.

To view the package history:

1. On the Deployments tab, click the name of the package whose history you want to view. The package’s details display on the screen.
2. Click **Version History** to display the versions in this package’s history. You can click a version to see a deployment report for that version, including which servers were deployed and any errors.



Package: My New Package

Following is the deployment history for this package. Click a deployment to see its details and any errors that might have occurred. To roll back to a previous version, click that version's Restore link.

Date	Last Action	Status	Applications	Servers	Restore
Wed Oct 07 00:28:36 GMT-700 2009	Deploy	Successful	notifications.war	Apache Tomcat 6.0.20-192.168.1.105	
Wed Oct 07 00:25:24 GMT-700 2009	Deploy	Failed	notifications.war	Apache Tomcat 6.0.20-192.168.1.105	Restore

Deployment Report

Server	Status	Errors
Apache Tomcat 6.0.20-192.168.1.105-8	Successful	
Apache Tomcat 6.0.20-192.168.1.105-8	Successful	

3. To restore a previous version, click the **Restore** link for that version. The link is available for successfully deployed versions only.
4. To return to a list of all packages, click **All** in the navigation pane on the left.

Redeploying a Package

If an error occurs, you can redeploy it by clicking **Redeploy**. Note that if you are currently modifying a package, clicking **Redeploy** will cancel your changes and redeploy the existing package. To save your changes, click **Save and Deploy** instead.

Deleting a Package

If you no longer need a package, you can delete it from the system. Deleting the package undeploys its applications from the target servers, but it does not delete those applications or servers from the repository. Note that if an application/target server combination is also in another package, you should redeploy that package to deploy that application to that server again.

To delete a package:

1. On the Deployments tab, click the name of the package you want to delete.
2. Click **Delete** to delete the package.
3. At the confirmation message, click **Yes** to delete the package and undeploy its applications from the servers specified in the package.

Monitoring a Server

This section describes how system administrators can use Tcat Server to monitor the health of a server, see which applications are up and which are down, and determine memory usage. To view server details, click the server name on the Servers tab. The Server Details screen displays the information on several different tabs, which are described in the sections below.

At any time when viewing the server details, you can click **Restart Server** at the top of the window to restart that server. If you no longer want to monitor the server, click **Unregister Server**.

Viewing and Editing the Server Summary

The Summary tab provides an overview of the server, including its status (displays a green light if it's running or a red light if it's stopped), the groups it belongs to, the packages that have been deployed to it, and its environment variables (see below).

To edit the summary details, click **Edit Server Info**, make your changes, and then click **Save**. To add the server to a group, select it from the drop-down list and click **Add**. To remove the server from a group, click the red X next to the group name.

Working with Environment Variables

By default, just the variables that have been set for the server via Tcat Server are displayed on the server's Summary tab. To display all environment variables on the server, click the **more** link. The full list of variables appears in a scrolling list below.

To manage the environment variables on the server, click **Edit Server Info**, and then do one or more of the following:

- To add an environment variable, type the variable's name and its value, and then click **Add**. As you type, matching variable names appear, allowing you to select one from the list.
- To change an existing variable, click in its name or value box and edit it as needed.
- To delete a variable, click the red X next to its value.

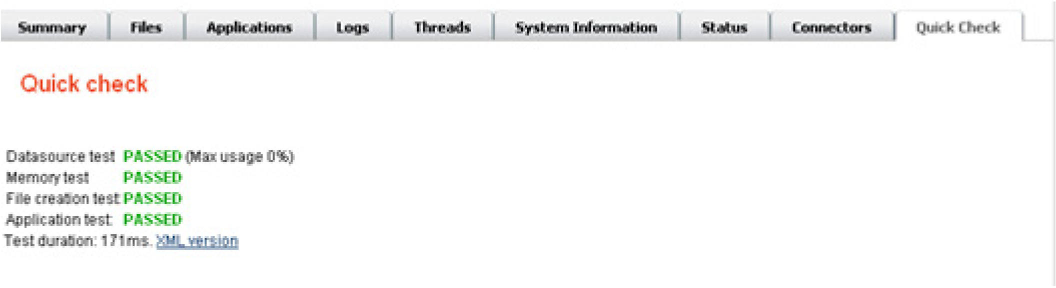
When you have finished making changes, click **Save**. To revert your changes, click **Cancel**.

If you change an environment variable, click **Restart Server** to restart the server and have your changes take effect.

Note If you make changes to your variables via Tcat Server, we recommend that you do not make further changes to the variables outside of Tcat Server. Otherwise, the variables will be out of sync, because they are pulled into Tcat Server from the JVM when Tomcat is started and are combined with any that you have set in the Tcat Server console. Therefore, if you must make changes outside of Tcat Server, also restart Tomcat manually outside of Tcat Server. The variables are then updated in the JVM and are pulled into Tcat Server, and any that you set explicitly in the Tcat Server console will take precedence over those set externally.

Performing a Quick Check

When viewing a server's details, you can get a quick overview of the server's health by clicking the Quick Check tab.



Quick check

```
Datasource test  PASSED (Max usage 0%)
Memory test      PASSED
File creation test PASSED
Application test PASSED
Test duration: 171ms. XML version
```

Clicking this tab runs four tests on the main areas of the system: data sources, memory, file creation, and applications. The quick check does the following:

- Scans all available data sources and generates a maximum usage score
- Allocates 1MB of memory into a byte array
- Creates and then deletes ten files in the temp directory

The check fails if any of the following is true:

- At least one of the declared data sources is 100% used
- Memory allocation test generated OutOfMemory exception
- File creation test encountered an IOException

The XML version allows you to run the quick check from automated application monitoring tools.

Working with the Server Configuration Files

The Files tab displays the files and folders on the server. You can view and edit any text-based (non-binary) file, redeploy it, and then restart the server.

To view a file, simply click its name. If you want to edit the file, click **Edit**, make your changes in the editor that appears, and then click **Save** at the bottom of the window. If you make a mistake, you can click **Reset** to reset the editor window to the original text in the file. To cancel your changes without saving and close the editor window, click **Cancel**.

To add a new file to the server, do the following:

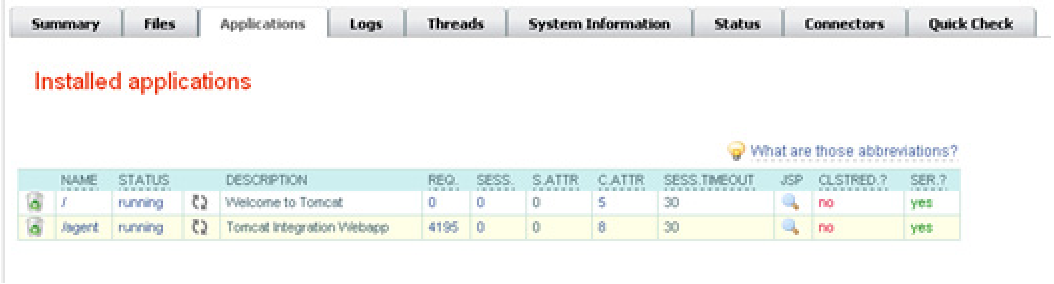
1. On the Files tab, click **New File**.
2. Enter the name you want to display for this file.
3. Specify the file (you can click **Browse** to select it from your file system) and click **Upload**.

To upload a new version of a file, click its name to view it, click **Upload New File**, specify the updated version of the file, and click **Upload**.

To delete a file, click its name to view it, and then click **Delete**.

Viewing the Deployed Applications

The Applications tab displays a list of the applications that have been deployed to this server. The Status column displays whether each application is currently running or stopped. You can stop or restart an application just by clicking its status to toggle it between running and down.



Installed applications

	NAME	STATUS		DESCRIPTION	REQ.	SESS.	S.ATTR	C.ATTR	SESS.TIMEOUT	JSP	CLSTRED.?	SER.?
🗑	/	running	⟳	Welcome to Tomcat	0	0	0	5	30	🔍	no	yes
🗑	/agent	running	⟳	Tomcat Integration Webapp	4195	0	0	8	30	🔍	no	yes

You can sort the list of applications by clicking any of the column headings. For example, you can click the Status column heading so that all stopped applications are grouped together at the top of the table.

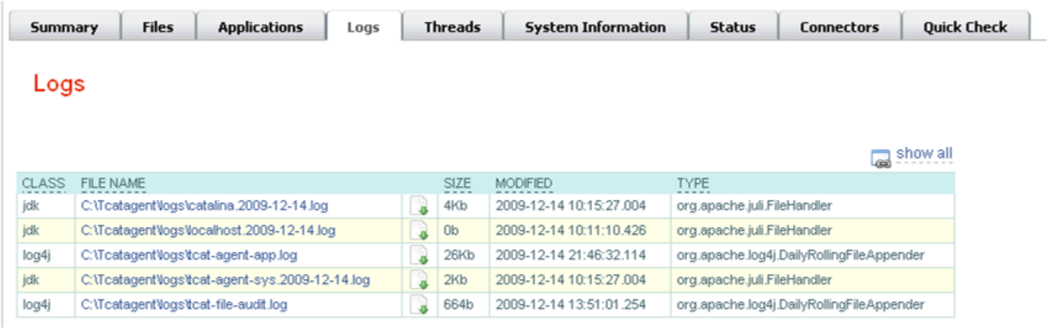
To view more details about an application, click the application name.

Following is a description of the columns:

Column	Description
Name	The name of the application folder.
Status	Whether the application is running or down (stopped).
Description	The description of the application as specified in the WAR file.
Req.	The total number of requests processed by each application.
Sess.	The total number of sessions for each application. A session is a connection between the application and a specific client, such as a user accessing the application through a web browser. Each session can process multiple requests. A session closes when the client terminates the session or when the session times out.
S.ATTR	The total number of session attributes. Session attributes store information about the client.
C.ATTR	The total number of ServletContext attributes set by this application. ServletContext attributes are shared by all servlets in the same application.
Sess. Timeout	The interval at which a session is automatically closed if there has been no activity within that time period.
JSP	Clicking the icon displays a list of all the JSPs in that application. You can then click a specific JSP to view its code. You can also compile one or more JSPs.
JDBC USAGE	Displays the maximum connection usage to JDBC resources by each application. You can click the bar in this column to see the JDBC resources for that application.
CLSTRED.?	Whether the application is distributable and deployed in a cluster.
SER.?	Whether all session attributes implement java.io.Serializable.

Viewing the Logs

The Logs tab displays the logs generated by Tomcat for each of the applications and for the Tomcat instance itself. Note that all logs are rolled over at midnight daily.



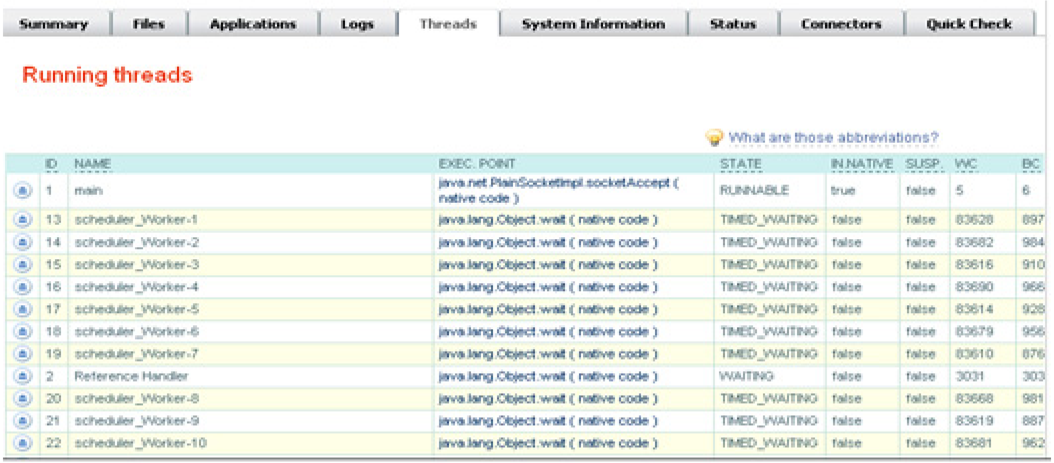
CLASS	FILE NAME		SIZE	MODIFIED	TYPE
jdk	C:\Tcatagent\logs\catalina.2009-12-14.log		4Kb	2009-12-14 10:15:27.004	org.apache.juli.FileHandler
jdk	C:\Tcatagent\logs\localhost.2009-12-14.log		0b	2009-12-14 10:11:10.426	org.apache.juli.FileHandler
log4j	C:\Tcatagent\logs\tcat-agent-app.log		26Kb	2009-12-14 21:46:32.114	org.apache.log4j.DailyRollingFileAppender
jdk	C:\Tcatagent\logs\tcat-agent-sys.2009-12-14.log		2Kb	2009-12-14 10:15:27.004	org.apache.juli.FileHandler
log4j	C:\Tcatagent\logs\tcat-file-audit.log		664b	2009-12-14 13:51:01.254	org.apache.log4j.DailyRollingFileAppender

The table displays the log type (e.g., JDK or Log4J), file name, file size, last-modified timestamp, and the class used to create the log. Click a log file name to view its contents, or click the download icon to save the file locally. When viewing the log file contents, you can use the zoom buttons to make the font larger or smaller, you can adjust the line wrapping, and you can clear the log file. The log is updated as it is being written; to pause it, click **Pause Tailing**. To resume watching it in real time, click **Resume Tailing**. To return to the list of logs, click **Back to log files list**.

For details on the logs generated by Tcat Server, see the *Tcat Server Administrator's Guide*.

Viewing the Threads

The Threads tab allows you to view the threads that are running in the JVM. The information displayed depends on whether the server's JVM has the JMX agent installed or is based on the thread class. The thread class view displays less-detailed information, so if you need more details, you should consider installing a JMX agent. To enable the JMX agent with Tomcat, add the `-Dcom.sun.management.jmxremote` option to your `CATALINA_OPTS` setting as described in "Installing Tcat Server" on page 21.



Running threads

	ID	NAME	EXEC. POINT	STATE	IN.NATIVE	SUSP.	WC	BC
	1	main	java.net.PlainSocketImpl.socketAccept (native code)	RUNNABLE	true	false	5	6
	13	scheduler_Worker-1	java.lang.Object.wait (native code)	TIMED_WAITING	false	false	83628	897
	14	scheduler_Worker-2	java.lang.Object.wait (native code)	TIMED_WAITING	false	false	83682	984
	15	scheduler_Worker-3	java.lang.Object.wait (native code)	TIMED_WAITING	false	false	83616	910
	16	scheduler_Worker-4	java.lang.Object.wait (native code)	TIMED_WAITING	false	false	83690	966
	17	scheduler_Worker-5	java.lang.Object.wait (native code)	TIMED_WAITING	false	false	83614	928
	18	scheduler_Worker-6	java.lang.Object.wait (native code)	TIMED_WAITING	false	false	83679	956
	19	scheduler_Worker-7	java.lang.Object.wait (native code)	TIMED_WAITING	false	false	83610	876
	2	Reference Handler	java.lang.Object.wait (native code)	WAITING	false	false	3031	303
	20	scheduler_Worker-8	java.lang.Object.wait (native code)	TIMED_WAITING	false	false	83668	981
	21	scheduler_Worker-9	java.lang.Object.wait (native code)	TIMED_WAITING	false	false	83619	887
	22	scheduler_Worker-10	java.lang.Object.wait (native code)	TIMED_WAITING	false	false	83681	962

The following tables describe the information displayed in each view.

JVM Agent Information

Column	Description
ID	A unique identifier for this thread. This value is assigned by the JVM whenever the application creates a thread.
NAME	The thread name. This value is assigned by the application.
EXEC. POINT	The point in the code where the thread was executed. The class name and line number are displayed if available.
STATE	The current state of the thread (see http://tinyurl.com/elank)
IN.NATIVE	Whether this thread is executing in native code.
SUSP.	Whether this thread is suspended (`Thread.suspend()` was called).
WC	The waited count (see http://tinyurl.com/le2m7b). This is the number of times the thread has been waiting.
BC	The blocked count (see http://tinyurl.com/ly4yxz). This is the number of times the thread has been blocked from entering a monitor. Typically, this happens when the thread has to wait when trying to enter a `synchronized()` block.

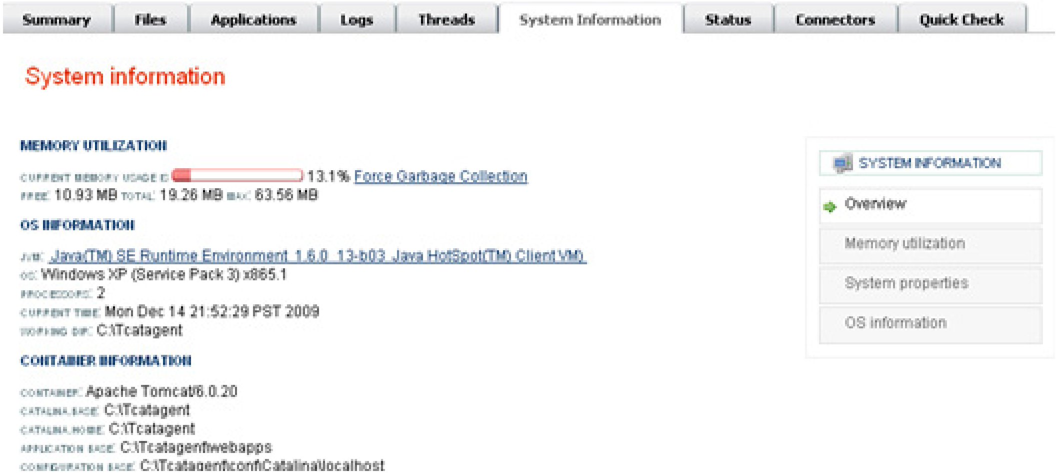
Thread Class Information

Column	Description
NAME	The thread name. This value is assigned by the application.
P	The priority of this thread. The higher the number, the higher its priority over other threads, and the more time it will get from the CPU.
APP	The application to which this thread belongs.
CLASS LOADER	The context class loader for this thread. Click the class to see the classpath the thread can access.
GROUP	The thread group to which this thread belongs.
THREAD CLASS	The class implementation that generated this thread.
RUNNABLE CLASS	The class that is executed by this thread.

Column	Description
D	Whether the thread is a daemon.
I	Whether the thread has been interrupted.

Viewing the System Information

The System Information tab displays information about the computer on which Tcat Server is installed.



System information

MEMORY UTILIZATION

CURRENT MEMORY USAGE: 13.1% Force Garbage Collection
FREE: 10.93 MB TOTAL: 19.26 MB MAX: 63.56 MB

OS INFORMATION

JVM: Java(TM) SE Runtime Environment 1.6.0_13-b03 Java HotSpot(TM) Client VM)
OS: Windows XP (Service Pack 3) x865.1
PROCESSORS: 2
CURRENT TIME: Mon Dec 14 21:52:29 PST 2009
WORKING DIR: C:\Tcatagent

CONTAINER INFORMATION

CONTAINER: Apache Tomcat/6.0.20
CATALINA.BASE: C:\Tcatagent
CATALINA.HOME: C:\Tcatagent
APPLICATION BASE: C:\Tcatagent\webapps
CONFIGURATION BASE: C:\Tcatagent\conf\Catalina\localhost

The memory utilization bar shows you at a glance how much memory is being used on the server. To release memory that's being taken up by objects that are no longer in use by the applications, click **Force Garbage Collection**. This link runs `System.gc()`, which advises the JVM to perform the garbage collection. Alternatively, you can click **Advise GC** (also runs `System.gc()`) or **Advise Finalization** (which runs `System.runFinalization()` to suggest running finalization methods on objects) on the Memory Utilization page. These options should only be invoked in a development or test environment, because they can cause a short pause in application execution and can possibly make the JVM freeze.

To view more specific information about memory utilization, click **Memory Utilization** in the vertical menu bar on the right side of the screen. The screen now displays memory usage for different caches and spaces and allows you to display a chart for each.

The rest of this tab displays information about the operating system and Tomcat container installed on this server. To display more specific information about the operating system, including available RAM and historical usage charts, click OS Information in the vertical menu bar on the right side of the screen.

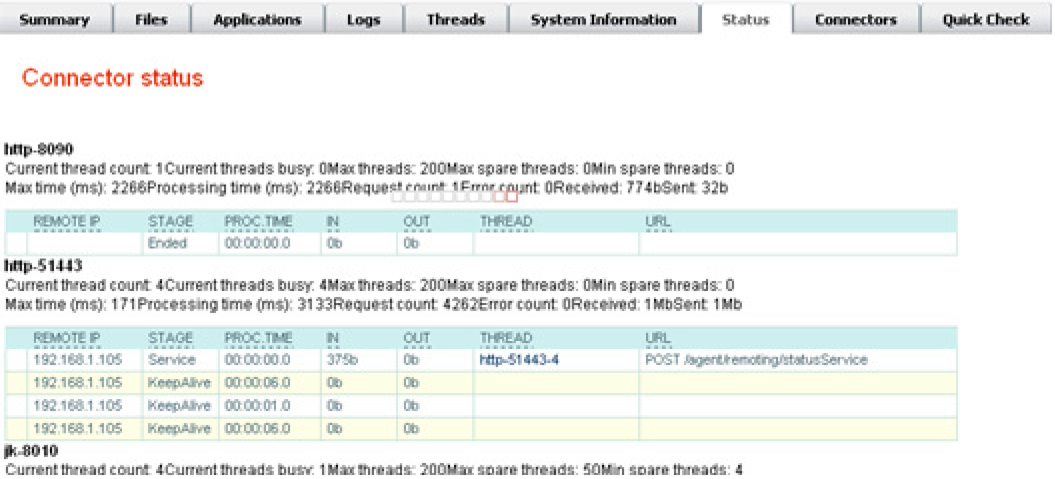
To view system properties, such as the Java home directory and Tomcat's base and home directories (`catalina.base` and `catalina.home`), click **System Properties** in the vertical menu bar.

Lastly, if your JVM is controlled by a Java Service Wrapper, you can click **Wrapper Control** to view information about the wrapper, including its Java PID, whether debug is enabled, and the values of all its properties.

Viewing the Connectors

There are two tabs that show you information about connectors: Status and Connectors.

The Status tab allows you to view the status of the connectors used to connect client requests to the applications.



Connector status

http-8090
Current thread count: 1Current threads busy: 0Max threads: 200Max spare threads: 0Min spare threads: 0
Max time (ms): 2266Processing time (ms): 2266Request count: 1Error count: 0Received: 774bSent: 32b

	REMOTE IP	STAGE	PROC.TIME	IN	OUT	THREAD	URL
		Ended	00:00:00.0	0b	0b		

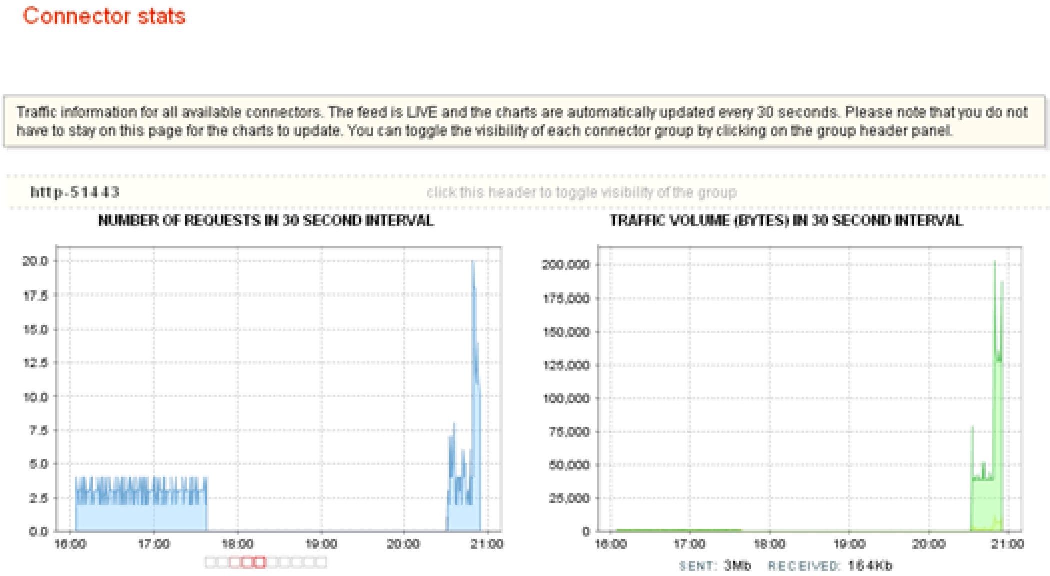
http-51443
Current thread count: 4Current threads busy: 4Max threads: 200Max spare threads: 0Min spare threads: 0
Max time (ms): 171Processing time (ms): 3133Request count: 4262Error count: 0Received: 1MbSent: 1Mb

	REMOTE IP	STAGE	PROC.TIME	IN	OUT	THREAD	URL
	192.168.1.105	Service	00:00:00.0	375b	0b	http-51443-4	POST /agent/remoting/statusService
	192.168.1.105	KeepAlive	00:00:06.0	0b	0b		
	192.168.1.105	KeepAlive	00:00:01.0	0b	0b		
	192.168.1.105	KeepAlive	00:00:06.0	0b	0b		

jk-8010
Current thread count: 4Current threads busy: 1Max threads: 200Max spare threads: 50Min spare threads: 4

The connectors are divided into groups based on the port and protocol they use. Each group displays information such as the total number of available and busy threads in that group. It also displays information for each individual connector, including the remote IP address, current stage, and the URL of the request.

The Connectors tab displays charts that show traffic volume information for all available connectors. The feed is live, and the charts are automatically updated every 30 seconds. You can toggle the visibility of a connector group by clicking that group's header panel.



Connector stats

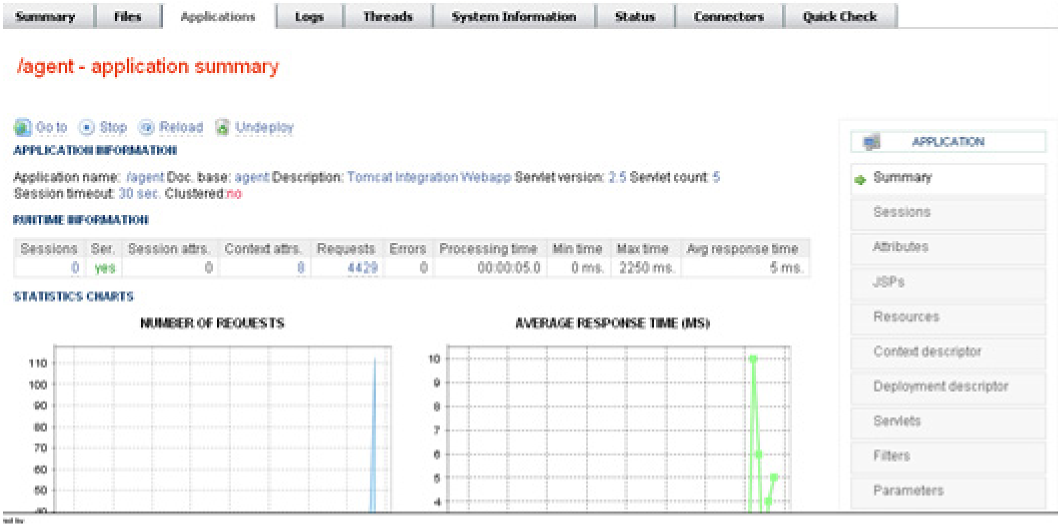
Traffic information for all available connectors. The feed is LIVE and the charts are automatically updated every 30 seconds. Please note that you do not have to stay on this page for the charts to update. You can toggle the visibility of each connector group by clicking on the group header panel.

http-51443 click this header to toggle visibility of the group

SENT: 3Mb RECEIVED: 164Kb

Monitoring Applications

When viewing a list of the applications on a server (in the Administration Console, click the Servers tab, click a server name, and then click the Applications tab), you can view further details about a specific application by clicking its name.



/agent - application summary

APPLICATION INFORMATION

Application name: /agent Doc. base: agent Description: Tomcat Integration Webapp Servlet version: 2.5 Servlet count: 5
Session timeout: 30 sec. Clustered:no

RUNTIME INFORMATION

Sessions	Ser.	Session attrs.	Context attrs.	Requests	Errors	Processing time	Min time	Max time	Avg response time
0	yes	0	8	4429	0	00:00:05.0	0 ms.	2250 ms.	5 ms.

The Summary view displays runtime information about the application and statistics charts. You can click one of the side tabs in the right navigation bar to see more information as described in the following sections.

Summary

The Summary view provides runtime information about the application, such as the number of servlets, context attributes, and requests. It also displays charts showing the historical number of requests and average response time. You can click a chart to expand it, and you can click any of the blue numbers to display more detailed information. For example, clicking the number of servlets displays the Servlets view. You can use the buttons at the top of the tab to go to the application (even if it’s on a remote server) in the browser window, to stop the application, to reload it, or to undeploy it from the server. You can also estimate the sessions size.

Sessions

The Sessions view displays the current sessions for this application. You can click the session name to see more details, including the session attributes, and to access the option to destroy the session. You can also click an IP address to perform a WHOIS query on it and display details about its origin. To estimate the size of the sessions, click **Estimate Size**. To hide the Size column, click **Hide Size**.

Attributes

The Attributes view displays the servlet context attributes for this application. If you want to remove an attribute from the servlet context, simply click the trash icon next to the attribute.

JSPs

The JSPs view lists the JSPs deployed with this application. If any have not been precompiled, you can select them and compile them from this view. To see the source code of a JSP, click its name. You can then click **View Generated Servlet** to view the auto-generated servlet code for this JSP or click **View JSP** to go back to the JSP code. You can also compile the JSP while viewing its code.

Resources

The Resources view lists the JNDI resources that are referenced by the application. The Type column displays the type of resource, and if it's a data source it also displays the URL, the maximum number of connections in the pool, the number of busy connections (cannot be reused), and the number of established connections (can be reused). The Link To column indicates whether this resource is a link on a globally defined resource (that is, a resource defined outside of the context that is shared by multiple applications). The Authentication column indicates whether resource authentication is handled by the container on behalf of the application (user name and password are specified on the resource definition) or programmatically by the application itself.

For information on working with JNDI resources in Tomcat, see "JNDI Resources How-To" on the Apache site (http://tinyurl.com/lh83d).

Context Descriptor

The Context Descriptor view displays the contents of the application's `context.xml` file. You can click the download link to download the file, which is renamed to the application's name.

Deployment Descriptor

The Deployment Descriptor view displays the contents of the application's `web.xml` file. You can click the download link to download the file.

Servlets

The Servlets view displays a list of the servlets in this application. For each servlet, the table displays the servlet's context path, the servlet class it implements, whether it's currently available, whether it's loaded on startup (and if so, its loading order), the time in milliseconds it took to load, the number of requests so far to this servlet, the total amount of time this servlet has spent processing requests, the number of errors, the minimum and maximum time spent on a single request, and, whether the servlet is multi-threaded. You can click **Servlet Mappings** to view the mappings between URLs and the servlets that handle them, and then click **Servlets** to return to viewing the list of servlets.

Filters

The Filters view displays a list of the filters used by this application. For each filter, the table displays the filter name, its implementation class, and a description (if available). You can click **Filter Mappings** to see the filters that are called for different URLs, which servlet implements the filter (if not specified by the application), which type of dispatchers apply the filter, and the filter class. Click **Filter Definitions** to return to viewing the list of filters.

Parameters

The Parameters view displays a list of the parameters that are used to initialize the context. Typically, these are taken from the application's `WEB-INF\web.xml` file.

Troubleshooting

This section describes common problems and how you can troubleshoot them.

What tools are available for troubleshooting?

You can use Tcat Server to check memory usage, see which objects are in your session and how big they are, view deployment descriptors, view threads, and view JSPs and their generated code. You can view these details by clicking a server on the Servers tab and then navigating through the tabs that show the server details. To see details for a particular application on the server, click the Applications tab when viewing the server details, and then click the application name.

For more information, see “Monitoring a Server” on page 38 and “Monitoring Applications” on page 49.

Problem: My applications are no longer recognized

When servers are registered, they are registered with a specific IP address. If you are testing on a laptop with a wireless connection and you move from one location to another, your IP address might change, and the servers will no longer be able to communicate with Tcat Server. In this case, you must unregister and reregister the servers with the current IP address.

Problem: My console is unavailable, so now I can't unregister its servers

To manually unregister a server, you must delete its `webapps\agent\WEB-INF\truststore.jks` file. The server will then become available again for registering in another console.

Problem: My newly installed Tcat Server instance is running, but it doesn’t appear in the Unregistered group in the console

If you are connected to a virtual private network (VPN), it can interfere with new Tcat Server instances appearing in the console. Disconnect from the VPN, restart the Tcat Server instance, and try again. You can verify that your Tcat Server agent is running by entering `http://`*`hostOrIPAddress`*`:`*`port`*`/agent` in your browser, such as `http://localhost:8080/agent` if it’s on your local computer. If you get a message indicating the status of the agent, it’s running.

After you have registered your servers, you can reconnect to your VPN.

Problem: I used Ctrl with the + key to increase the font size, and the screen went white

If you are using Firefox, and you press the **Ctrl** key with the plus (**+**) key to increase the magnification of the screen, some of the screens can become blank. To work around this issue, press **Ctrl** plus the minus key (**-**) to zoom back out.

Problem: I want to change which files can be edited from the Files tab when viewing a specific server's details

The Files tab provides you with access to files on the local server as well as remote servers. To restrict or enable write access to a directory on a specific server, you modify the `agent/WEB-INF/spring-services.xml` file on that server and add or remove directories from the `writeExcludes` property in the `accessControlFileService` bean.

Chapter 4
Deploying to Amazon EC2

This chapter describes how to use the Tcat Server EC2 plug-in to deploy new EC2 images and register Tomcat instances on them, all from within the Tcat Server administration console. It includes the following sections:

Installing the Plug-in

To install the EC2 plug-in, simply download it from the Tcat Server download page (http://tinyurl.com/m4b2pr) and copy it to your *TCAT_HOME*`/webapps/console/WEB-INF/plugins` directory. This directory must be owned and have the same permissions as TOMCAT_USER. When you run the administration console, the Amazon EC2 tab will appear.

If you are using Mule iBeans, you must delete the `mule-ibeans/lib/modules/deployed/ibeans-module-spring-1.0-beta-9-full.jar` to enable iBeans and the EC2 plugin to work in the same instance of Tcat Server.

Creating an Amazon EC2 Account

Before you can get started, you must create an Amazon EC2 account at http://aws.amazon.com/ec2/. Note your access key ID and secret key. If you have an existing account, you can find them by logging into the EC2 web site, clicking **Your Account**, and selecting **Security Credentials**.

You will also need to create a key pair if you do not already have one. To do this, log into the EC2 Console (https://console.aws.amazon.com/ec2), click **Key-Pairs** on the left, click **Create Key Pair**, and then enter a key pair name.

Opening Ports

By default, Amazon creates a firewall that prevents any communication from happening between the outside world and your EC2 instances. To change this, you must download the EC2 tools from http://tinyurl.com/c2blwj so you can open ports. You then issue the following commands:

```
$ ec2-authorize -p 8080 default
$ ec2-authorize -p 51433 default
```

This will open up ports 8080 and 51433 by default.

Amazon Machine Images

Amazon Machine Images (AMIs) are images that get provisioned to each EC2 instance. You should familiarize yourself with the AMI concepts in the EC2 documentation before proceeding.

You will want to create your own AMI (see page 57) with a Tcat Server instance on it. For testing purposes, we provide an image with Tcat Server and Ubuntu 9.04 from Alestic (http://alestic.com/). On startup, this image will start Tcat Server and make it available on port 8080. Please note that we provide this image on an unsupported basis solely for testing, and the image ID and details of this image may change in the future. The test image ID is: ami-f7d8389e

Using the Plug-in

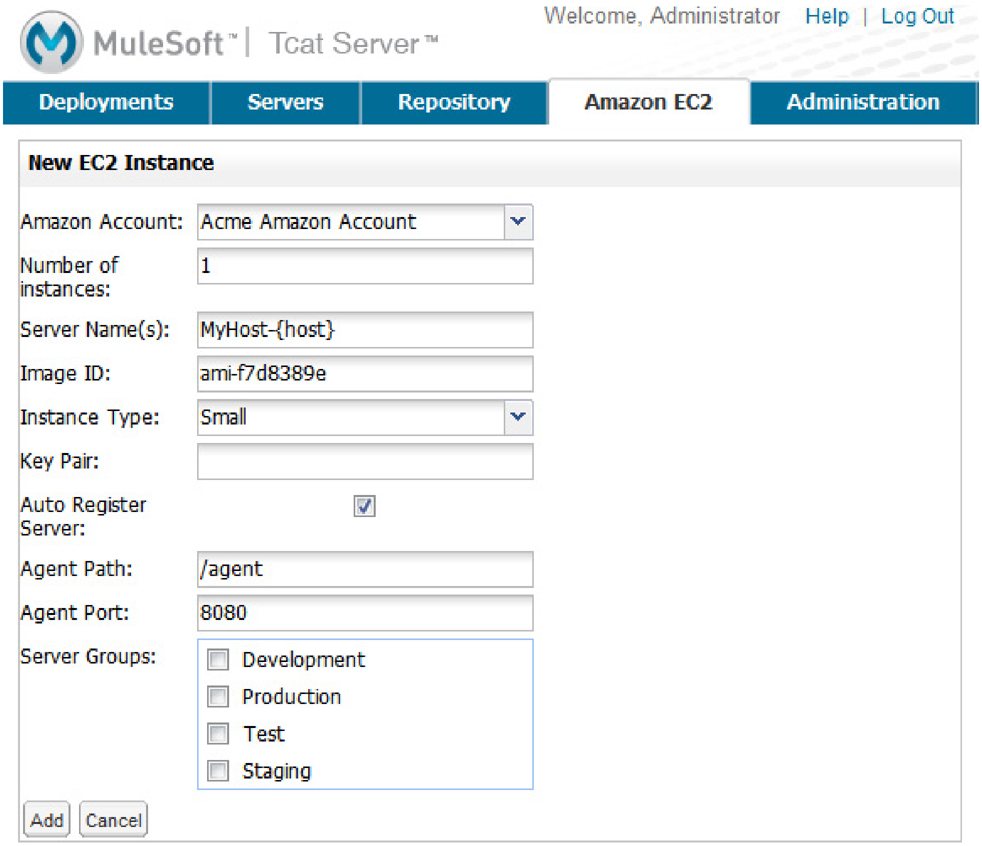
The Amazon EC2 tab in the Tcat Server administration console allows you to manage your EC2 accounts and instances. The first step is to create the account, and then you create your instances.

To create the account:

1 On the Amazon EC2 tab, click **New Account**.

2 Enter a name for the account, your Amazon access key ID, your secret key, and the name of the key pair you created.

3 Click **OK**.



New EC2 Instance

Amazon Account:	Acme Amazon Account
Number of instances:	1
Server Name(s):	MyHost-{host}
Image ID:	ami-f7d8389e
Instance Type:	Small
Key Pair:	
Auto Register Server:	☑
Agent Path:	/agent
Agent Port:	8080
Server Groups:	☐ Development ☐ Production ☐ Test ☐ Staging

‣ To create an instance:

1. On the Amazon EC2 tab, click **New Instance**.
2. Select the Amazon account you created.
3. Enter the number of EC2 instances you want to create.
4. Enter the name of the server (instance). If you are creating multiple instances, you can use the `{host}` variable to insert the name of the host into the instance name, such as `MyHost-{host}`.
5. Enter the ID of the AMI you created, or use the test AMI ID (see "Amazon Machine Images" on page 55).
6. Specify the instance type (size). This will affect how much your Amazon account is charged.
7. Enter the name of the key pair to use with these servers.
8. If you want to automatically register these instances with your Tcat Server administration console, so that you can manage these Tcat Server instances, select the Auto Register Server check box.
9. Enter the URL path where the Tcat Server agent WAR will reside. By default, set this to `/agent`.
10. Enter the port to use for the Tcat Server agent.
11. Select any server groups you want to add these new servers to.
12. Click **Add**.

After the plug-in issues a new instance request, it takes a while for Amazon to provision the image. The instances list will display `pending` until the image is created, at which point it will display `running`. Your new Tcat Server instance will then be available on the Servers tab if you opted to automatically register it.

Creating a Tcat Server AMI

If you are deploying Tcat Server to a production environment, typically you create your own AMI with Tcat Server on it. The following Amazon resources provide information on creating AMIs:

- Creating an AMI: http://tinyurl.com/ydgn4qt
- Official AMI database: http://tinyurl.com/6y347e

Configuring Tcat Server for Automatic Startup

To ensure that you can automatically register your new Tcat Server instance, you should configure the Tcat Server instance on your AMI to start automatically on server startup. On Linux, you can do this through an init script as follows.

1. Install Tcat Server (or Tomcat and the Tcat Server agent WAR) into your `/usr/local/tcat` directory.
2. Download the init script from http://tinyurl.com/y8qqy99 and copy it to `/etc/init.d/tomcat`.
3. Execute `chmod +x /etc/init.d/tomcat`
4. Execute `chmod +x /usr/local/tcat/conf/tomcat-env.sh`
5. Execute the `update-rc.d tomcat defaults` command.

   This will configure Ubuntu to run the init script on startup.
6. If you install Tcat Server in a different location, edit the `/etc/init.d/tomcat` script APP_ENV variable to reflect where Tcat Server is installed.

Index

MuleSoft™

201 Mission Street
Suite 510
San Francisco, CA 94105
Phone: 877-MULE-OSS (877-685-3677)
Fax: 415-227-0842
www.mulesoft.com