



Tcat Server Administrator's Guide

Version 6 R4
October 2010

Confidential

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

Copyright

Copyright © 2003-2010, MuleSoft, Inc. All rights reserved. No part of this publication may be copied or distributed, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, manual, optical, chemical or otherwise; or disclosed to third parties without the express written permission of MuleSoft, Inc.

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

MuleSoft, Mule and MuleForge are among the trademarks of MuleSoft. All other product names are trademarks of their respective companies.

Part number: 6.2en_us2009.12.8

Table of Contents

Preface	8
Who Should Read This Guide?	8
Typographic Conventions	8
Technical Support	9
 Chapter 1	
Introduction to Tcat Server	10
What is Tcat Server?	10
About Tomcat	11
 Chapter 2	
Getting Started	12
Quick Start	12
Installing Tcat Server on One Machine	12
Installing Tcat Server on Multiple Machines	17
Upgrading to Tcat Server 6 R4	20
Run the Installer	20
What the Installer Does	20
Installing Multiple Tcat Server Instances on a Single Machine	21
Installing Tomcat Separately	25
Starting and Stopping Tcat Server	26
Starting and Stopping on Windows and Linux	27
Additional Options on Windows	27
Additional Options on Linux	27
Starting and Stopping on Mac OS X	28
Starting the Administration Console	28
Modifying JAVA_OPTS	28
Installation Customization Options	29
Make Contents of Webapps Directory Unwritable	29
Renaming the tc6 Service on Linux	30
Adding Tcat Server Capabilities to an Existing Apache Tomcat Installation	31

Add Tcat to Existing Tomcat Install	31
Converting a Stock Tomcat Windows Service Into a Tcat Windows Service	32
Add Tcat agent.war to Tomcat Installation Directory	34
Add New Server by IP Address	35
Moving a Tcat Installation from a Non-Linux OS to Linux	36
Uninstalling Tcat Server	38
Upgrading From Earlier Versions to Tcat Server 6 R2	38
Upgrading via the Installer	38
LDAP Upgrade	39
Amazon EC2 Plugin Upgrade	39

Chapter 3

Working with Servers and Applications 40

Tcat Server Global Dashboard	40
Customizing the Dashboard	41
Server Dashboard	42
Server Metric Portlet	43
Working with Servers	44
Registering a Server	45
Using a Server Profile	45
Restarting a Server	46
Creating a Server Group	46
Renaming a Group	47
Finding a Server	47
Editing Server Information	47
Adding Servers to a Group	49
Removing Servers from a Group	49
Deleting a Server Group	50
Unregistering a Server	50
Manually Adding a Server	51
Deploying Applications	51
Creating a Deployment	51
Uploading Applications	52
Modifying a Deployment	53
Viewing History on a Deployment	54
Monitoring a Server	55
Viewing and Editing the Server Summary	55
Working with the Server Configuration Files	56
Viewing the Deployed Applications	57
Viewing the Logs	58

Viewing the Threads	59
Viewing the System Information	61
Viewing the Connectors	62
Monitoring Applications	63
Summary	63
Sessions	64
Attributes	64
JSPs	64
Resources	64
Context Descriptor	65
Deployment Descriptor	65
Servlets	65
Filters	65
Parameters	65
Working With Alerts	66
Overview of Alerts	66
Viewing Raised Alerts	66
Defining Alerts	69
Defining Notifications and Destinations	75
Troubleshooting	85

Chapter 4

Managing Cloudcat Instances	89
Prerequisites	89
Purchase a Subscription to Cloudcat	90
Creating a Cloudcat Instance	90
Using Tcat Server to Manage Cloudcat Instances	91
Deploying Tomcat Applications to Cloudcat	92
Getting Support for Cloudcat	92

Chapter 5

Deploying to Amazon EC2	93
Installing the Plug-in	93
Creating an Amazon EC2 Account	94
Opening Ports	94
Amazon Machine Images	94
Using the Plug-in	95
Creating a Tcat Server AMI	96
Configuring Tcat Server for Automatic Startup	97

Chapter 6**Working with the Repository 98**

Viewing the Repository Contents.	98
Modifying an Artifact or Workspace.	98
Adding a Workspace.	99
Adding Applications and Other Artifacts	99
Subscribing to Change Notifications	100
Setting Security	100
Removing an Artifact or Workspace.	101

Chapter 7**Extending Tcat Server 102**

Automating Tasks.	102
Creating a Script.	102
Modifying a Script	103
Scheduling a Script	103
Cron Command Syntax	104
Scripting Examples.	106
Redeploying a Named Package	106
Automatically Redeploying After an Artifact Update.	106
Sending Notifications of New Artifact Versions.	108
Replicating Workspaces.	110
Using the REST API	110

Chapter 8**Administering Tcat Server 112**

Managing Users and Groups	112
Managing Users	112
Managing User Groups.	114
Enabling Authentication Through LDAP	115
Set up Tcat Server Roles	116
Add the LDAP JAR to Your Classpath	116
Create the LDAP Configuration File.	116
Working with Server Profiles	118
Creating Remote Workspaces.	120
Monitoring Repository Activity	120
Applying Your Tcat Server License	121
Scheduling the Usage Statistics Script.	121
Working with Logs.	122
File Audit Log.	122

Shell Script Log.	123
Appendix A	
Using the Maven Publishing Plug-in	125
Overview	125
Adding the Tcat Server Repository to the POM.	125
Configuring the Plug-in	126
Configuring Security Information	127
Additional Configuration Options	127
Index	129

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 <code>server.xml</code>
<i>AaBbCc123</i>	Placeholder text that you change.	<code>http://serverName:8080/console</code>
AaBbCc123	A live link to a web site, email address, or another section in the document	See page 7
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

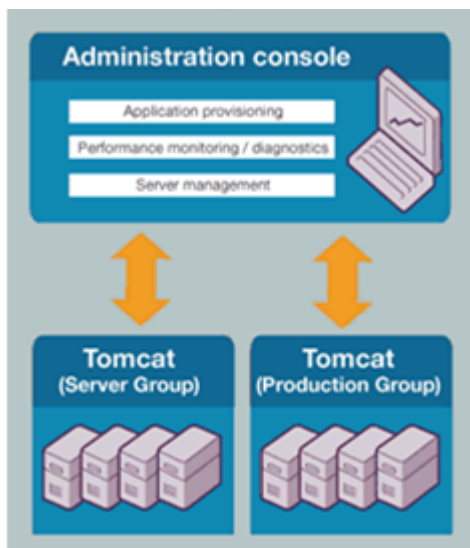
Introduction to Tcat Server

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

- “**What is Tcat Server?**” on page 9
- “**About Tomcat**” on page 10

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 and an administration console.



Tcat Server makes Tomcat even more powerful and simple to use:

- Administration Console web application enables you to monitor and control all your Tomcat (and Tcat Server) 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.
- Install multiple instances on a single machine
- Run Tcat Server instances as Windows Services
- Restart remotely (except for OS X)
- Use Admin Shell component to run custom groovy scripts, hook into deployment events, and automate restarts

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.

About Tomcat

Tomcat is a very popular web server and servlet container developed by [Apache Software Foundation][<http://www.apache.org>]. 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 7 Information: <http://www.tomcat7.com/>
- 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, including installation instructions.
- **“Upgrading to Tcat Server 6 R4”** on page 19
- **“Installing Multiple Tcat Server Instances on a Single Machine”** on page 20
- **“Starting and Stopping Tcat Server”** on page 25
- **“Modifying JAVA_OPTS”** on page 27
- **“Installation Customization Options”** on page 28
- **“Adding Tcat Server Capabilities to an Existing Apache Tomcat Installation”** on page 30
- **“Uninstalling Tcat Server”** on page 37

Quick Start

Use these instructions to install Tcat Server on one or more machines, and to get started managing, monitoring, and deploying your applications.

- **“Installing Tcat Server on One Machine”** on page 11
- **“Installing Tcat Server on Multiple Machines”** on page 16

Installing Tcat Server on One Machine

This process explains how to install the Tcat Server and the Tcat Administration Console on one machine, then use the Console to manage the Server and deploy a new application.

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). 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 its alias, the JRE_HOME environment variable, is set.

Red Hat Enterprise Linux / CentOS / Fedora

Before installing Tcat Server on a 64-bit installation, you must have the 32-bit compatibility libraries and all of their dependencies installed. You may install them via yum like this for most versions of RHEL, CentOS, and Fedora:

```
$ sudo -s
# yum install compat-libstdc++-33.i586 gtk+.i586 gtk2.i586
# exit
```

Or, like this for some newer versions of these distributions:

```
$ sudo -s
# yum install compat-libstdc++-33.i686 gtk+.i686 gtk2.i686
# exit
```

Then, you may proceed with installing Tcat Server via the installer executable.

Ubuntu Linux

Before installing Tcat Server on a 64-bit installation of Ubuntu, you must have the 32-bit compatibility libraries and all of their dependencies installed. You may install them via apt-get like this:

```
$ sudo -s
# sudo apt-get install ia32-libs
# exit
```

Then, you may proceed with installing Tcat Server via the installer executable.

OpenSUSE Linux

Before installing Tcat Server on a 64-bit installation of OpenSUSE 11, you must have the 32-bit versions of GTK2 and all of its dependencies installed. You may install them using yast, like this:

```
$ sudo -s
# yast -i gtk2-32bit
# exit
```

Then, you may proceed with installing Tcat Server via the installer executable.

Automated Installation

The Tcat Server installers are available for Windows, Linux, and Mac OS X. These installers allow you to install Tomcat, the Tcat Server agent, the Tcat Server administration console, and each installs a preconfigured version of Tomcat, saving you some manual steps.

Headless Installation for Windows

To run the Tcat Server 6.3 Installer using all defaults (including the Administration Console and using all the default ports), execute the following command on a command prompt:

```
start /wait tcat-installer-6.3.0.exe -c
```

Step One: Run the Installer

Run the installer and select Standard Install, then accept all defaults:

Installer Executables

Windows	Linux	OS X
installer.exe	installer.sh	installer.dmg

Step Two: Start Tcat Server

Use the commands listed below to start Tcat Server for your platform.

Tcat Server Platform Start Commands

Windows	Linux	OS X
tc6.bat OR start menu	./tc6.sh	./startup.sh

Step Three: Logon to the Tcat Server Console

- 1 After a minute or two, open a Web browser to the following address:
`http://localhost:8080/console`

- 2 Log in using "admin" for both username and password.

The Tcat Server console appears, with the Servers tab selected by default.

Step Four: Register the Server with the Console

- 1 In the left panel, click Unregistered.

The local Tcat server instance appears to the right.

- 2 Click the checkbox next to Server Name.

- 3 Click Register.

The local server instance now appears in the All group with a green status indicator

Step Five: Manage the Tcat Instance

- 1 Click the All list in the left panel.

- 2 Click on Server Name.

The Server Details panel appears.

- 3 Click through the tabs to view the information and operations available.

Step Six: Add a New Application

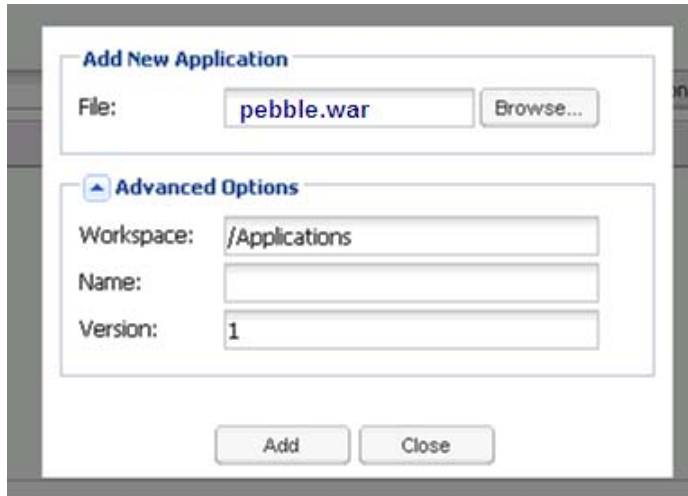
- 1 Click the Deployments tab.

- 2 Click on New Package at top right.

- 3 Click Upload New Application.

- 4 Browse to local file system to select a .war file.

- 5 Click Add, and enter Package Name.



6 Click in the Space to the top right, and select the server name.

7 Click Save and Deploy at bottom right.

8 Select Deployed from left panel.

The new package appears under Applications to the left, with the server to the right.

9 Verify your application works by navigating to it in a browser.



Installing Tcat Server on Multiple Machines

This section explains how to install the Tcat Server and the Tcat Administration Console on one machine, to install only the Tcat Server on another machine, and then use the Administration Console to manage servers and applications on both Tcat Server instances.

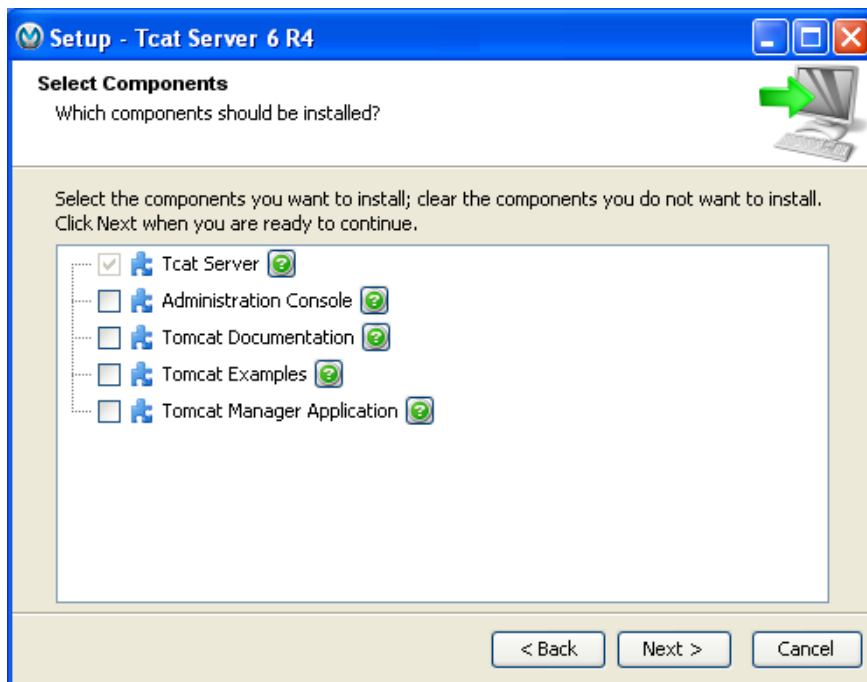
Note If you are connected to your network via VPN, you should disconnect before you continue. Otherwise, the Tcat Server instances may not be automatically detected.

Step One: Install Tcat Server on One Machine

See “[Installing Tcat Server on Multiple Machines](#)” on page 16 for instructions.

Step Two: Install Tcat Server Without the Console

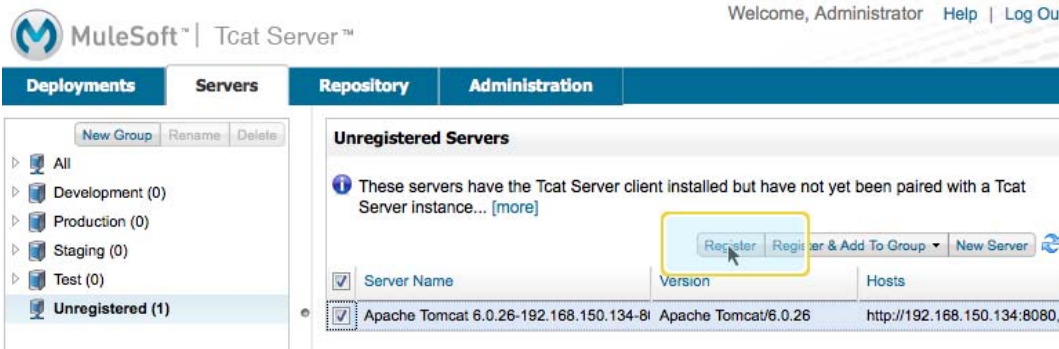
- 1 Run the installer again on the other machine(s), selecting Custom Installation.
- 2 De-select Administration Console.



- 3 Start the server.

Step Three: Pair New Tomcat Server to Administration Console

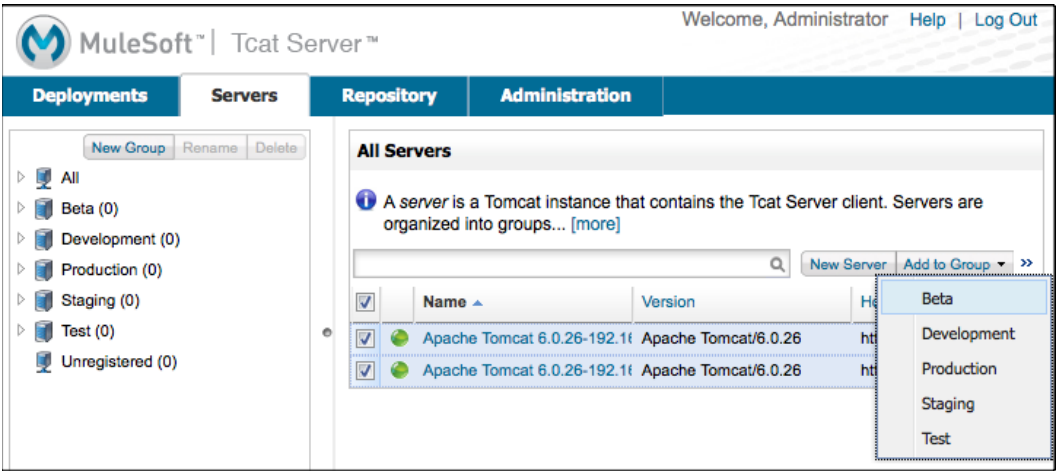
- 1 In the Administration Console of the first server installed, click on the Servers tab.
The newly installed instance appears in the Unregistered group
- 2 Click the checkbox next to Server Name.
- 3 Click Register.



The local server instance now appears in the All group with a green status indicator.

Step Four: Add Server to Group

- 1 From the Servers tab in the Administration Console, click New Group.
- 2 Enter "Beta" and click OK.
- 3 From the list of all servers which is displayed, select both server instances.
- 4 Click Add to Group, and select the Beta group from the dropdown menu.



- 5 Confirm your selection.

Both instances appear in the Beta group.

Upgrading to Tcat Server 6 R4

If you have Tcat Server R3 installed, the new installer will retain all paired servers, new web applications, user roles, server profiles, and environment variables you have saved.

If you are using an earlier version of Tcat Server, you need to follow the upgrade path to R3 first.

Download the Tcat Server installer for Windows (.exe), Linux and Solaris (.sh), and Mac OS (.dmg) from <http://www.mulesoft.com/download-tcat-server-enterprise-tomcat>.

Run the Installer

- 1 Run the installer.

Installer Executables

Windows	Linux	OS X
installer.exe	installer.sh	installer.dmg

- 2 Point to the existing application install directory.

The installer retains all your existing settings..

What the Installer Does

The Tcat Server 6 R4 installer is designed to perform the following tasks:

- 1 Back up the agent web application (copies webapp/agent to tcat-backup/agent) and unpack the new one.
- 2 Back up the console web application (copies webapp/console to tcat-backup/console) and unpack the new one.
- 3 Back up any user-modified Tomcat configuration files in the conf directory tree (copies conf to tcat-backup/conf).

- 4 Preserve old agent.properties which defines which port the agent is listening on.
- 5 Preserve agent truststore.jks used to provide security settings associated with the agent

Once the installer has completed the upgrade, you should compare the new Tcat Server configuration files in the conf directory with the older configuration files in `tcat-backup/conf` to determine which of your changes you want to carry forward into the new Tcat Server configuration.

Installing Multiple Tcat Server Instances on a Single Machine

This procedure enables you to run multiple instances of Tcat Server on a single machine, using one instance of the Tcat Administration Console (or more) to manage multiple instances of Tcat Server, each running within their own JVM.

Install Tcat Server with Administration Console

This process explains how to install multiple Tcat Server instances on one machine, then use the Administration Console to manage, monitor, and deploy servers and applications.

Before You Begin

For this procedure, you need a working installation of Java JDK 1.5 (not a JRE) or better.

Ensure that the `JAVA_HOME` environment variable is set correctly.

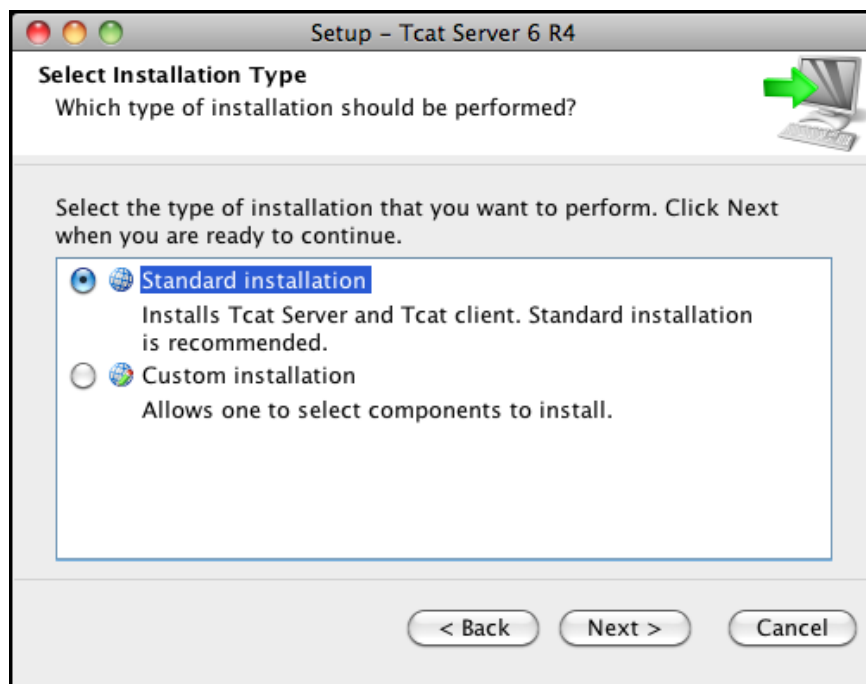
Download the Tcat Server installer for Windows (.exe), Linux (.sh), and MacOS (.dmg) from <http://www.mulesoft.com/download-tcat-server-enterprise-tomcat>.

Step One: Run Installer, Accepting All Defaults

Select Standard Install, and accept all defaults.

Installers

Windows	Linux	OS X
installer.exe	installer.sh	installer.dmg



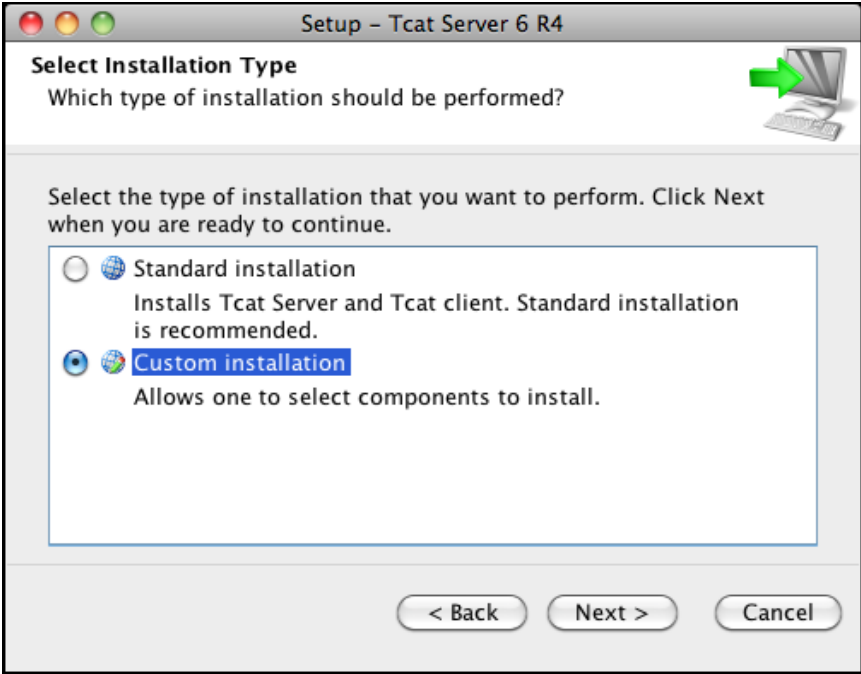
Note After the first install, start the server and leave it running.

Install Additional Tcat Servers

In this step, you will run the installer again, selecting Custom and determining the ports for HTTP, HTTPS redirect, Tcat Server Shutdown and AJP connector port.

Run the Tcat Installer with Options

Perform the following steps for each additional instance you want to have locally installed.



- 1 Run the installer. At Select Installation Type, choose Custom Installation.
- 2 At the next window, deselect Administration Console from the list.
- 3 At the Port Settings page, enter unique port numbers according to the following table:

Port Settings

Port	Default	Location
HTTP/1.1	8080	conf/server.xml
redirectPort	8443	conf/server.xml
Server SHUTDOWN	8005	conf/server.xml
AJP/1.3 Connector	8009	conf/server.xml
securePort	51443	conf/Catalina/localhost/agent.properties

Startup Tcat Server Instances

Start all the Tcat Server instances. (The the one with the Administration Console should already be running.)

Startup Tcat Server Instances

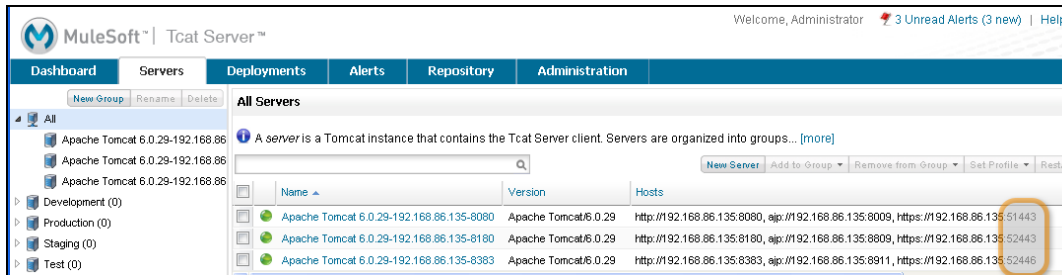
Windows	Linux	OS X
bin/tcat6 start	as administrator: sudo service tcat6 start	catalina.sh start
	as non-administrator: cd /opt/TcatServer6/bin/tcat6 start	

Login, Pair, and Manage Tcat Servers

- Login to Administration Console
 - 1 After a minute or two, open a Web browser to the following address:
`http://localhost:8080/console`
 - 2 Log in using "admin" for both username and password
 - 3 The Tcat Server console appears, with the Servers tab selected by default
- Pair Servers
 - 1 1. In the left panel, click Unregistered
 - 2 The local Tcat Server instance appears to the right
 - 3 Click the checkbox next to Server Name
 - 4 Click Register
 - 5 The local server instance now appears in the All group with a green status indicator

Repeat for each Tcat Server instance you installed.
- Manage Servers
 - 1 Click the All list in the left panel
 - 2 Click on Server Name
 - 3 The Server Details panel appears
 - 4 Click through the tabs to view the information and operations available

5 Click on Server Summary to verify secure port numbers are distinct.



Rename Servers

- 1 From the Servers tab in the Administration Console, click on one Tcat Server instance
- 2 The Server Summary appears
- 3 Click Edit Server Info
- 4 Enter a unique name in Server Name
- 5 Click Save
- 6 The new server name appears immediately in the list of Servers

Rules About securePort Setting

- 1 Any port number from 1 to 65535 if the Tomcat JVM runs as root. Any number from 1024 to 65535 if the Tomcat JVM runs as a user other than root.
- 2 Must not conflict with any of the other TCP ports being used by Tomcat and/or webapps running in the same JVM that have opened server sockets.

Note On non-Windows OSs, ports 1-1023 inclusive are "privileged ports" and require root privileges to bind to them. Those are the only special port numbers, and only on non-Windows OSs.

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.

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 26
- “**Additional Options on Windows**” on page 26
- “**Additional Options on Linux**” on page 26
- “**Starting and Stopping on Mac OS X**” on page 27
- “**Starting the Administration Console**” on page 27

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 27](#).

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 27](#).

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 27](#).

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 43](#).

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`

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 54) 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*).

Installation Customization Options

This section describes some procedures for customizing installs.

Make Contents of Webapps Directory Unwriteable

By default, the Administration Console enables a user to edit files on any Tcat Server instance registered to it.

This property is set in the `spring-servlets.xml` file in the `webapps/agent/WEB-INF/` directory: Below is the relevant snippet:

```
<property name="writeExcludes">
  <list>
    <value>lib/catalina*.jar</value>
    <value>**/tomcat*.jar</value>
    <value>conf/tcat-overrides.conf</value>
    <!-- block the webapps directory -->
    <!-- <value>webapps/**</value> -->
  </list>
</property>
```

To disable this ability, take the following steps:

- 1 Uncomment last element shown in the above snippet, replacing the following

```
<!-- <value>webapps/**</value> -->
```

with the following:

```
<value>webapps/**</value>
```

- 2 Save the file
- 3 Restart the Tcat Server instance

Renaming the tcat6 Service on Linux

You may wish to rename your Tcat Server's init script, either because you're installing more than one copy of Tcat Server in a single operating system and you need to prevent an init script naming conflict, or because you want to invoke the init script using a different name. Tcat Server supports renaming the service.

First, make sure you shut down your Tcat/Tomcat instance whose service you want to rename, as follows:

```
$ sudo service tcat6 stop
```

Or, if you're currently using a stock Tomcat package init script:

```
$ sudo service tomcat6 stop
```

Switch to a root shell:

```
$ sudo -s
```

Set the new service name as an environment variable, along with the absolute path to the directory to the Tcat Server installation you're changing the service name for:

```
# export NEW_SERVICE_NAME=t1
# export TCAT_HOME=/opt/TcatServer6
```

Next, rename the init script symlinks to the new service name (*copy and paste these commands; do not type them in*):

```
# mv /etc/init.d/tcat6 /etc/init.d/$NEW_SERVICE_NAME 2>/dev/null
# mv $TCAT_HOME/bin/tcat6 $TCAT_HOME/bin/$NEW_SERVICE_NAME 2>/dev/null
# mv $TCAT_HOME/conf/Catalina/localhost/tcat6
$TCAT_HOME/conf/Catalina/localhost/$NEW_SERVICE_NAME
```

If any of the above "tcat6" files do not exist, it is because you installed Tcat Server's agent webapp only, which is okay. You must pair the agent with the console before the agent will unpack its service scripts.

And, in your Tcat/Tomcat instance's environment file, which is used for the JVM's startup environment, change the service name setting (*copy and paste these commands; do not type them in*):

```
# sed -i .bak -e "s/\-Dtc\.\ service\=[I nstal lation^ ]*  
/-Dtc\.\ service=$NEW_SERVI CE_NAME /g" \  
$TCAT_HOME/conf/Catal i na/ l ocal host/tcat-env. conf
```

Exit from the root shell.

```
# exi t
```

If you're changing the service in order to install two or more Tcat Server installations in a single operating system, you should also ensure that the port numbers in Tomcat's <tomcatHome>/conf/server.xml do not conflict, and also that the Tcat Server agent secure port number of each Tcat Server instance is unique (copy and paste these commands – don't type them in):

```
# export NEW_AGENT_SECURE_PORT=51444  
# sed -i .bak -e "s/^securePort=. */securePort=$NEW_AGENT_SECURE_PORT/g" \  
$TCAT_HOME/webapps/agent/WEB-INF/agent. properti es
```

Then inspect the agent.properties file to ensure the setting is correct.

The default agent secure port is 51443.

You're now finished renaming the service. You can now start, stop, or restart Tcat Server using the service name you chose:

```
$ sudo service t1 start
```

Adding Tcat Server Capabilities to an Existing Apache Tomcat Installation

This section describes how to add Tcat Server Monitoring and Management capabilities to an existing instance of the open source Tomcat server.

Add Tcat to Existing Tomcat Install

This process explains how to install the Tcat Server on an existing Tomcat server instance, and to use the Administration Console to register, manage, and monitor that server.

This process requires one or more existing Tomcat instances, as well as an installed instance of Tcat Server with the Administration Console on the same machine or on a different machine. If you use more than one machine, these machines must have TCP network connectivity from the Console to the Tomcat machines you want to manage. The Console must be able to open initial connections over either HTTP (8080 by default) or AJP (8009 by default), and then over the Tcat Agent secure port (51443 by default). All of these port numbers are configurable.

See the supported environments table below:

Supported Environments

Requirements	Environments Supported
Tomcat 5.5 or higher	Any operating system that has a fully compliant Java 1.5 or 1.6 JDK or JRE

Note For instructions on installing Tcat Server with the Administration Console, consult the QuickStart. Instructions on installing Tomcat server are at <http://tomcat.apache.org>.

Converting a Stock Tomcat Windows Service Into a Tcat Windows Service

The following applies to Tcat Server versions 6.4.1 and 6.4.0 only.

This procedure assumes that you're converting only a single installation of stock Tomcat per machine to a Tcat Server, but if you'd like to run more than one converted Tomcat installation per server you will need to rename each installation's service name. The default Tcat windows service name is "TcatServer". Wherever you're seeing "TcatServer" in the below instructions, replace that string with a different service name, and also change `conf/Catalina/localhost/tcat-env.conf` to contain your modified service name. The file `tcat-env.conf` will only exist after the first successful server registration with the Tcat Server console.

- First, unregister your stock Tomcat from your Tcat Server console, if you had previously installed the agent webapp and registered it. Make sure to remove the agent's `truststore.jks` file from your Tomcat's `webapps/agent/WEB-INF/` directory, if it exists, then restart your Tomcat.

- Install a full Tcat Server installation (Windows service) on a different machine, so that you can copy some of its individual files into your stock Tomcat Windows service installation directory tree. You do not need to start it – just install it to unpack the files. Make sure the Tcat Server that you install is version "Tcat 6 R4" (6.4.1 or 6.4.0).
- Copy the following files from the Tcat installation to the Tomcat installation:
 - TcatServer6\bin\tcat6.bat to Tomcat6\bin\tcat6.bat
 - TcatServer6\bin\tcat-service.bat to Tomcat6\bin\tcat-service.bat
 - TcatServer6\bin\catalina.bat to Tomcat6\bin\catalina.bat
 - TcatServer6\bin\prunsrv64.exe to Tomcat6\bin\prunsrv64.exe
 - TcatServer6\bin\prunsrv32.exe to Tomcat6\bin\prunsrv32.exe
 - TcatServer6\bin\prunmgr.exe to Tomcat6\bin\TcatServerw.exe

Modify the service name in the destination filename if you are customizing the service name.

- Open an administrator shell and change directory into your Tomcat's CATALINA_HOME directory.
- Make sure that the JAVA_HOME or JRE_HOME (not both) environment variable is set properly to your choice of Java VM in the current shell environment. Take extra care to verify that it is set exactly right.
- Run the Tcat Windows service installer, passing the Windows service name you want to use for this Tcat Server installation, like this:

```
Tomcat 6>cd bin
```

```
Tomcat 6\bin>tcat-service install TcatServer
```

Carefully inspect the last line of the output of this command. If it results in an error such as "Failed installing 'TcatServer' service", then it is likely that you already have a service with that name installed, and you'll need to perform the above steps over with a different service name.

- Copy the Tcat Server agent.war webapp archive into your Tomcat's webapps/ directory, assuming you have not already done so.
- Restart your Tomcat via the stock Tomcat windows service.
- Once your Tomcat has restarted, use the Tcat Server console to register your Tomcat server. Do not restart it from the console yet!

- In your stock Tomcat's installation tree, edit the file `conf/Catalina/localhost/tcat-env.conf`. If you are naming this server's Tcat service name something other than the default "TcatServer" name, then change all occurrences of: `-Dtc.service=TcatServer` to your customized service name. Don't modify any other settings yet.
- Shut down your stock Tomcat windows service.
- Start your Tcat Server windows service by running this from your admin shell:

```
Tomcat 6\bin>tcat6 start
```

Or, if you're customizing the service name, give your modified service name as a second argument:

```
Tomcat 6\bin>tcat6 start MyServiceName
```

That's it. Your Tomcat is now running as a Tcat Server windows service.

From the Tcat console, the server should be available. Inspect it. Try a Tcat Server restart. Click the Restart Server button for this server. This should restart the new Tcat Server windows service, and bring it back up. From this point onwards, use only the Tcat Server console to restart your Tomcat server, or use the Tcat Server windows service (per the Tcat documentation), if you need to start, restart, or stop your Tomcat locally without the Tcat Server console. You should disable the stock Tomcat windows service by moving the `bin\tomcat6.exe` file to a different filename so that it is not accidentally used.

Also, from this point onwards, make any environment variable changes only from the Tcat Server console or the Windows system properties control. Environment variable changes you make in the console will get saved into your `tcat-env.conf` file, and will be used during Tcat Server restarts.

Add Tcat agent.war to Tomcat Installation Directory

Follow these instructions if you are activating the Tcat agent in an installation of Tomcat that is not installed as a Windows service. This includes all operating systems.

Copy the `agent.war` file from the `TcatServer6/webapps` directory of the Tcat Server installation into the `webapps` directory in your Apache Tomcat installation, e.g. `C:\Program Files\apache-tomcat-6.0.26\webapps`.

A few seconds after the Tcat agent web application starts on this server, you should be able to see it as a new Tomcat instance in the Unregistered Group of the Tcat Server Administration Console.

Add New Server by IP Address

If the newly modified Tomcat server does not appear in the Administration Console, click on the New Server button to the upper right of the group list.

- 1 Enter a Server Name and the Tcat Agent Url, click Add

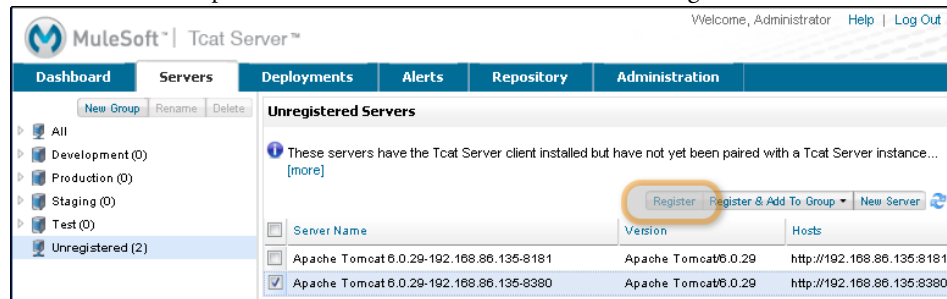


The screenshot shows the MuleSoft Tcat Server Administration Console. The 'Servers' tab is selected, and the 'Unregistered (3)' group is highlighted in the left sidebar. The 'Add Server' dialog is open, prompting the user to enter a server name and a Tcat Agent URL. The 'Server Name' field contains 'XP VM Apache Tomcat' and the 'Tcat Agent URL' field contains 'http://192.168.86.131:8080/agent'. There are 'Add' and 'Cancel' buttons at the bottom of the dialog.

The new server should now appear in the All Group of the Administration Console.

Register the Unregistered Tomcat Server

- 1 From the Administration Console, click on the Unregistered Group
- 2 Select the new Apache Tomcat server from the list and click Register



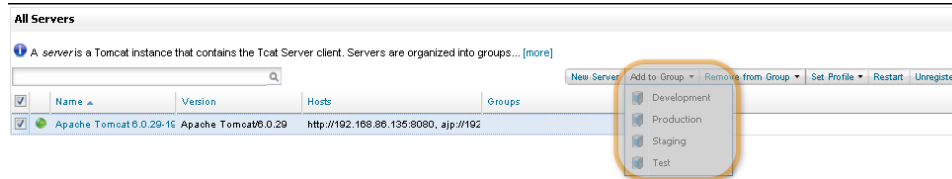
The screenshot shows the MuleSoft Tcat Server Administration Console with the 'Unregistered Servers' tab selected. A message states: 'These servers have the Tcat Server client installed but have not yet been paired with a Tcat Server instance... [more]'. Below this is a table of unregistered servers. The 'Register' button is highlighted with a red circle. The 'New Server' button is also visible.

Server Name	Version	Hosts
Apache Tomcat 6.0.29-192.168.86.135-8181	Apache Tomcat 6.0.29	http://192.168.86.135:8181
<input checked="" type="checkbox"/> Apache Tomcat 6.0.29-192.168.86.135-8380	Apache Tomcat 6.0.29	http://192.168.86.135:8380

Add Server to Group

This procedure is optional.

- 1 From the Administration Console, select the server
- 2 Click Add to Group and select the group from the pull down menu



- 3 Confirm the choice and wait for the Administration Console to refresh

The server appears in the new group.

Moving a Tcat Installation from a Non-Linux OS to Linux

This page discusses how to move an already-installed Tcat Server installation directory tree from a server running an operating system other than Linux to another server machine that runs Linux. This information applies to the following Linux distributions: Red Hat, CentOS, Fedora, Ubuntu, Debian, Debian variants, and SUSE variants.

Obviously, a whole-machine migration of server software won't necessarily be easy to do. If it is a simple Tcat installation where not much was changed versus the stock Tcat Server installation, and the webapps are part of Tcat's \$CATALINA_HOME tree, then you can recursively copy that Tcat installation tree across to the Linux machine, and all you would need to do afterwards is (expect that you need to be root for all of this):

- 1 If you run Tcat on a Java runtime that didn't come with Tcat, you should install the same Java brand and version runtime on the destination Linux server. It is best to use exactly the same release version of Java on the destination machine as you had on the original machine.
- 2 Before changing anything else, unregister the server you're moving from Tcat's console.
- 3 Shut down the instance you're about to move.
- 4 Recursively copy all of your files to the destination Linux machine. Preferably, you should attempt to preserve file ownerships, groups, and permissions, in the process of copying the installation tree.

- 5 Make sure that you create (or verify the existence of) the user and group you want to run the Tcat JVM as. We suggest you use the same user and group names that you were using on the previous machine.
- 6 Make sure that the owner and group of the files makes sense for the user you're going to run it as on Linux. Usually, we run it as user 'tomcat', where the group can be either 'tomcat' or 'root'. You may want to use these commands to change the owner ID and the group ID:

```
# cd /opt/TcatServer6
# find . -uid 46 -exec chown tomcat {} \;
# find . -gid 5000 -exec chgrp tomcat {} \;
```

- 7 You should symlink the Linux init script into place, if you have root / administrator privileges, like this:

```
# ln -s /opt/TcatServer6/bin/tcat6 /etc/init.d/tcat6
```

- 8 You should tell Linux to start the tcat6 service on reboots. On Red Hat, CentOS, Fedora, and SUSE / openSUSE:

```
# chkconfig tcat6 --level 2345 on
```

On Ubuntu, Debian, and Debian variants:

```
# update-rc.d tcat6 defaults
```

- 9 Edit your Tcat's conf/Catalina/localhost/tcat-env.conf file, if there is one, and make sure that all of the paths and the settings are valid for Linux. If this proves to be too difficult, you can alternatively delete or move this file out of the way.
- 10 Make sure to configure other back-end machines to allow communicating with the destination Linux server. This includes any back-end machines or servers that the deployed webapps must connect to or communicate with, such as SMTP servers, database servers, and messaging servers.
- 11 Make sure that the webapps/agent/WEB-INF/truststore.jks file does not exist. If it exists, delete it.
- 12 Delete or move the log files from the logs/ directory so that Tcat will write fresh logs on the destination Linux machine.
- 13 At this point, Tcat should run fine on Linux. Try starting it. Look through the log files and verify that there are no unexpected exceptions.
- 14 From the Tcat console, register the server instance.
- 15 Try triggering a server restart from the Tcat console – this should succeed.

That's it. The Tcat installation is now migrated to the Linux server machine.

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.

Upgrading From Earlier Versions 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.

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 5, “[Deploying to Amazon EC2](#).”

Chapter 3

Working with Servers and Applications

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

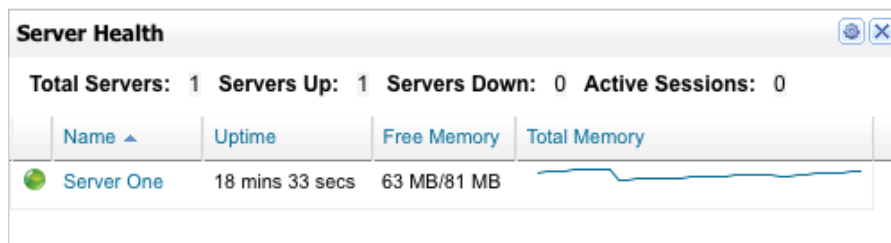
- “**Tcat Server Global Dashboard**” on page 39
- “**Working with Servers**” on page 43
- “**Deploying Applications**” on page 50
- “**Monitoring a Server**” on page 54
- “**Monitoring Applications**” on page 62
- “**Working With Alerts**” on page 65
- “**Troubleshooting**” on page 84

Tcat Server Global Dashboard

The Tcat Global Dashboard allows you to create a unique view into your Tcat Server instance via the provided customizable portlets.

By default, Tcat Server includes the Server Health, Application Status, Raised Alert and Deployment Status portlet—and many more are currently under development!

- **Server Health Portlet**—Allows you to monitor the health and key memory metrics of specific servers or just servers within a specific group. It also includes a sparkline graph of memory usage trends.



- **Application Status Portlet**—Allows you to monitor the availability and number of active sessions of the web applications contained in your deployments.

Webapp Status			
<div>Start Stop Reload</div>			
Name ▲	Deployment	Status	Sessions
/portal	customer portal	0 / 0 / 0	0


- **Deployment Status Portlet**—Allows you to monitor the state of your deployments, or just the deployments in defined server groups.

Deployment Status			
Deployments: 1 Failed: 0 Undeployed: 1 Unreconciled: 1			
Name ▲	Status	Webapps	Last Modified
⚠ customer portal	Undeployed	portal.war [20100920-1	Mon Sep 20 10:03:00 PDT 2010

- **Raised Alerts Portlet**—Allows you to quickly see what alerts, if any, have occurred in your Tcat Server instance. You can also limit this to specific server groups.

Raised Alerts		
Total Alerts: 1 Unread: 0 Fatal: 0 Critical: 1		
Server	Name	Date
⬆ Spongebob	Server Up	Mon Sep 20 10:14:40 PDT 2010

Customizing the Dashboard

You can rearrange the position of each portlet on the dashboard by "drag and dropping" the portlet to the desired location. Portlets can be removed by clicking on the close  icon in the top right corner of each portlet. Adding a new portlet is as simple as selecting the desired type from the drop down "Add Portlet" menu.

If a portlet supports customization you can do so by clicking on the configure icon in the top right portlet container—this will render the configuration panel which will vary depending on available options for each portlet type.

Raised Alerts

Portlet Name:

My Important Alerts

Groups:

Select a Group...

Add

Development

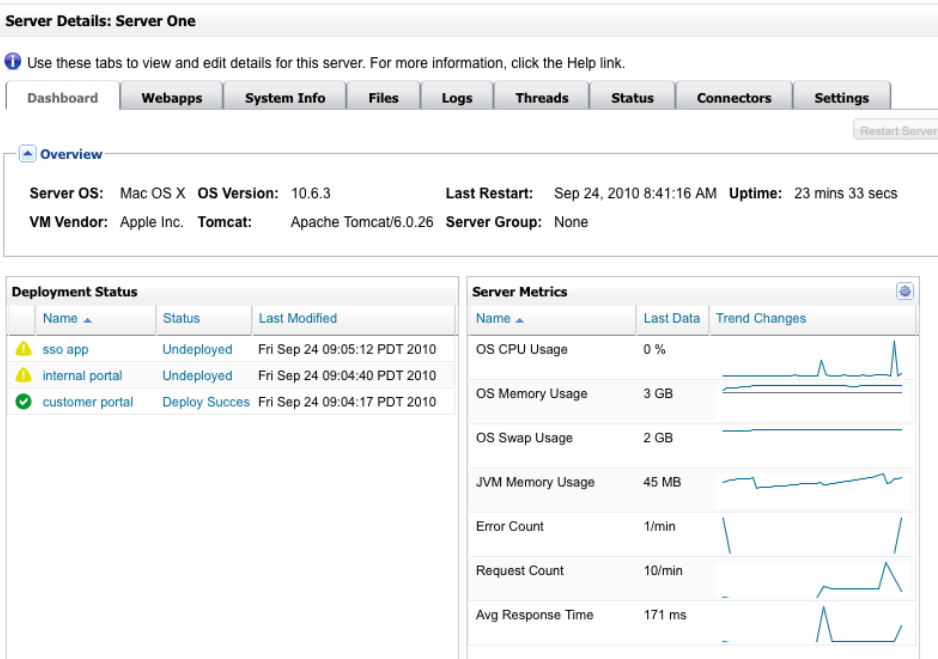
Production

Save

Cancel

Server Dashboard

The following figure illustrates the Tcat Server dashboard.



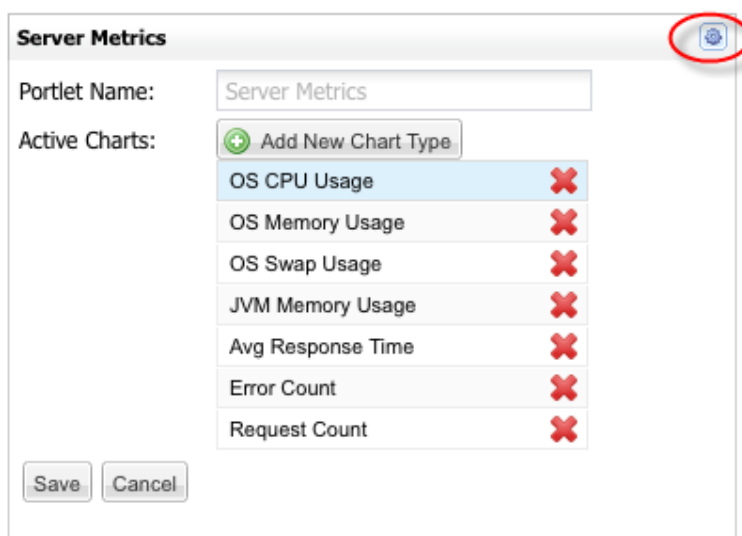
Server Metric Portlet

The Server Metric Portlet allows you to graph predefined metrics and trends—as well as user-defined JMX objects and attributes.

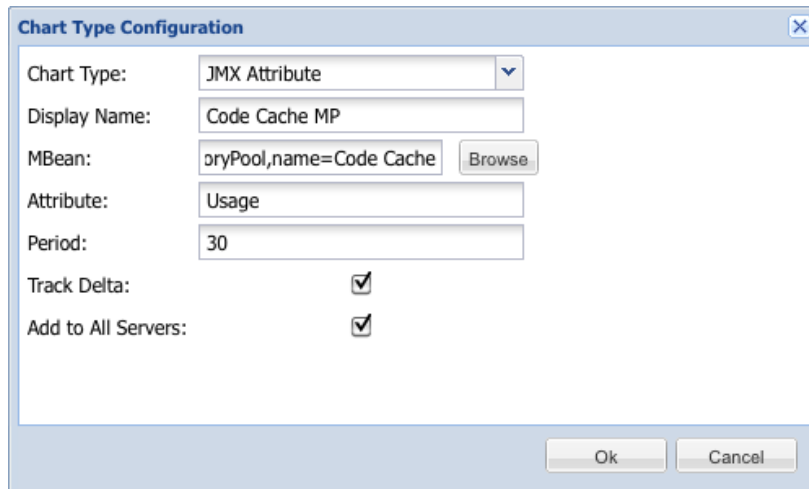
The complete list of available chart types includes:

- OS Swap Usage
- Error Count
- Average Response Time
- JMX Attributes
- Request Count
- OS Memory Usage
- JVM Memory Usage
- OS CPU Usage

To configure the elements rendered on this portlet select the gear icon in the top right corner.



To add a new chart type, select the "Add New Chart Type" button. You should then see another configuration panel. Depending on the type you select, you'll need to fill in the required form fields. Select OK. You should then see the item you just added in the list. Remember to select Save to retain your changes.



The image shows a 'Chart Type Configuration' dialog box with the following fields and options:

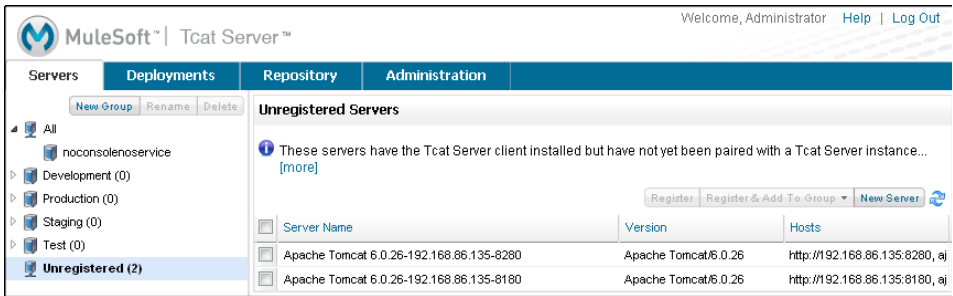
- Chart Type: JMX Attribute (dropdown menu)
- Display Name: Code Cache MP (text field)
- MBean: jryPool,name=Code Cache (text field) with a 'Browse' button
- Attribute: Usage (text field)
- Period: 30 (text field)
- Track Delta: ☒
- Add to All Servers: ☒
- Buttons: Ok, Cancel

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.



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 50).

- ▶ **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

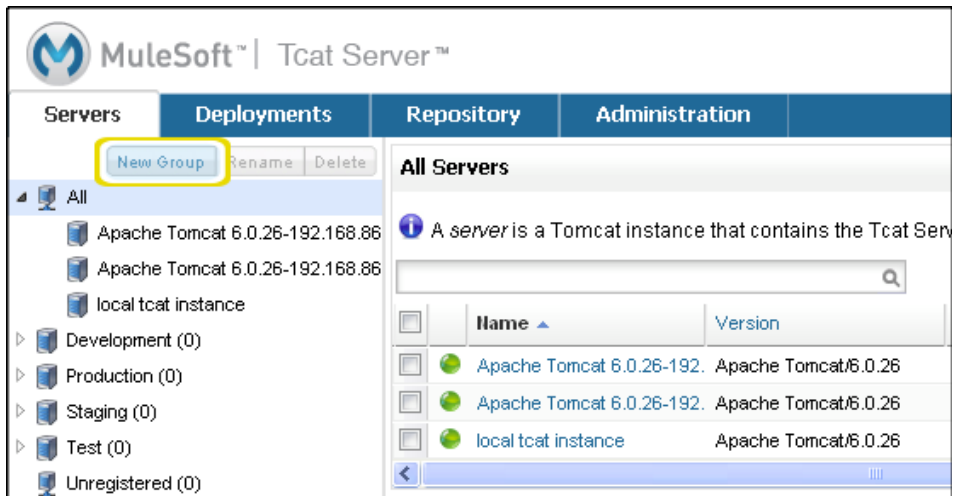
You can restart one or more servers by selecting their check boxes and clicking **Restart**. You can also restart a server when you are viewing its details.

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 44).

► To create a server group:

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



- 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 50), 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

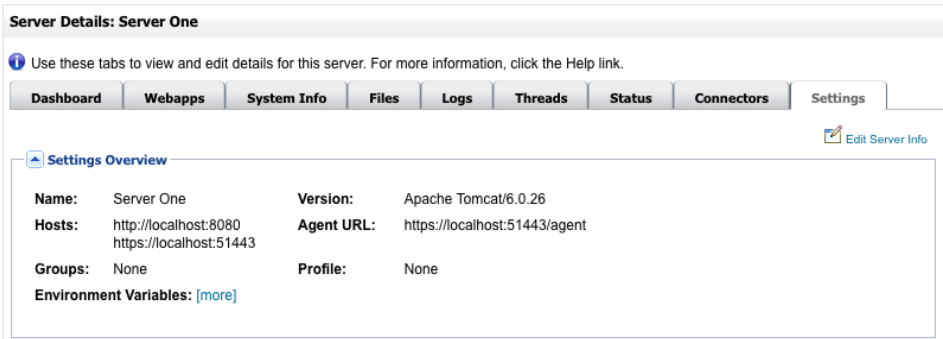
There are several ways to find a server in the Tcat Server Administration Console:

- 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**.

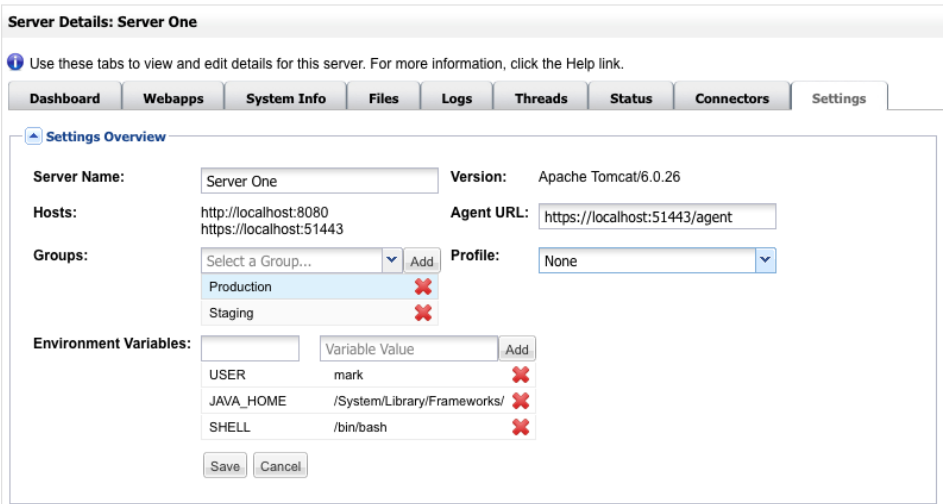
Editing Server Information

After a Tcat Server instance is registered with the Administration Console, you can edit server information with the following steps:

- 1 Select the Tcat Server instance from the Servers tab.
- 2 Select the Settings tab.



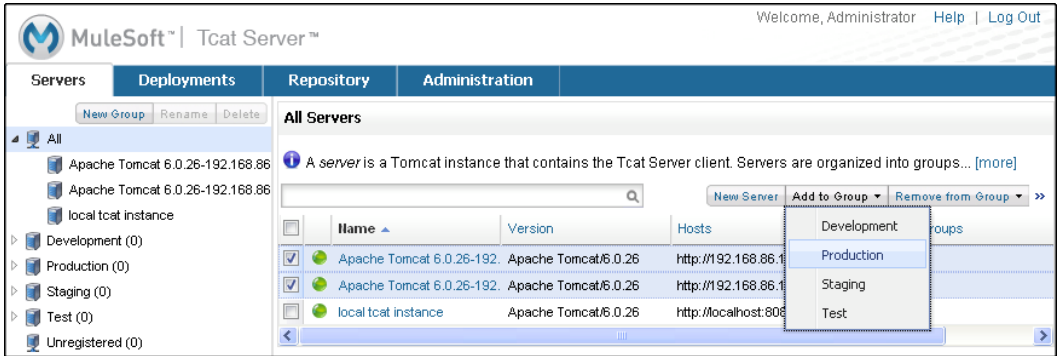
- 1 Click Edit Server Info:
 - 1 Server Name
 - 2 Groups
 - 3 Profile
 - 4 Agent URL
 - 5 Environment Variables
- 2 Click Save



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.



- 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://local host: 8080/agent` (replacing `local host` and `8080` with the correct host and port).

Deploying Applications

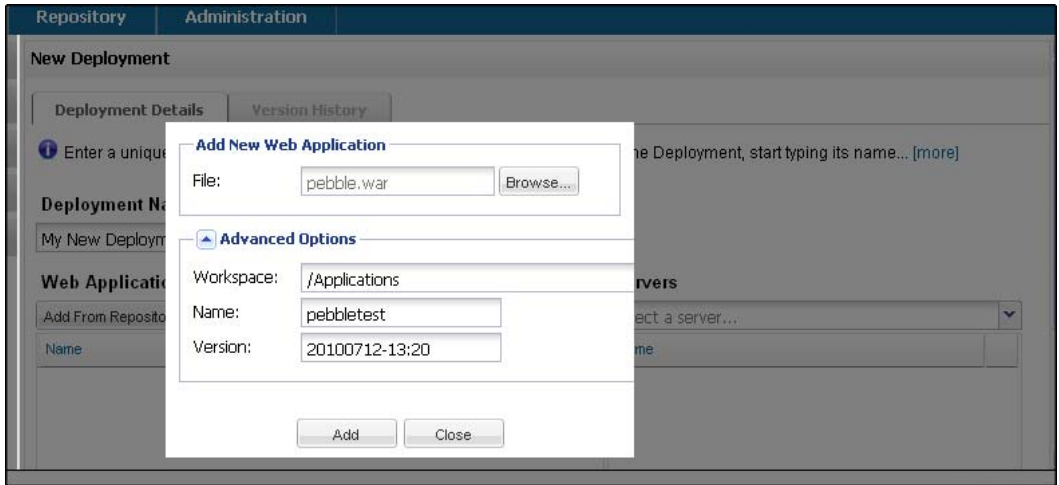
Deployment is the mechanism that enables you to deploy one or more applications to multiple Tomcat instances or groups, and to undeploy them just as easily. This page describes the various tasks related to deployment.

Creating a Deployment

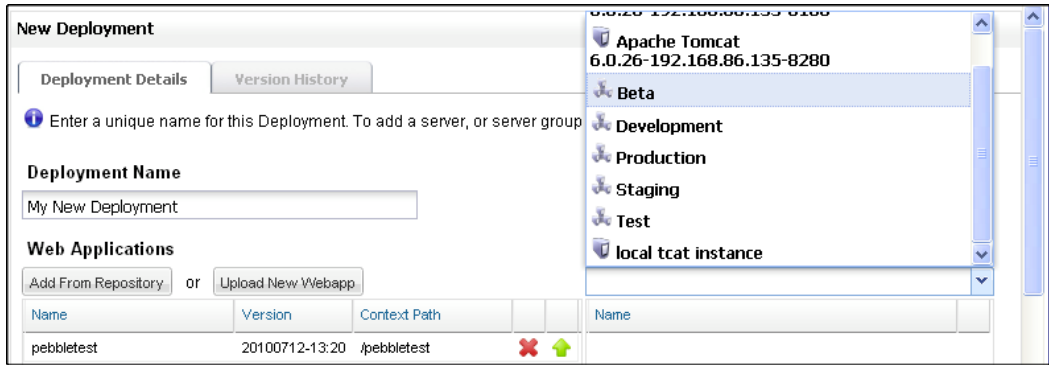
A Deployment is a grouping of applications you can deploy to servers or groups.

► **To create the deployment:**

- 1** On the Deployments tab, click New Deployment and enter a unique name for this entity
- 2** Just beneath the Web Applications list, clic Add from Repository or Upload New Version
- 3** Browse to the .war file and click Open
- 4** For a new web application, you can edit the URL context path by clicking Advanced Options and entering it in the Name field
- 5** Edit version information (optional)
- 6** Click Add



- 7 To the right of the Web Applications list, select the target server or group for this deployment



- 8 To the bottom right, click Save

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 52](#)).

- Add applications directly into the repository using the Repository tab (administrator permissions are required).
 - 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.

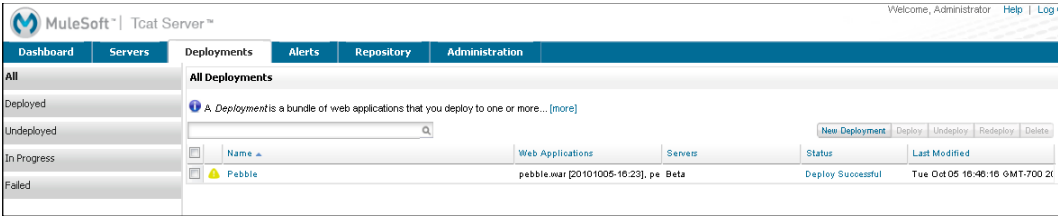
For information on managing an application, such as moving it from one location to another in the repository, see the *Tcat Server Administrator's Guide*.

Modifying a Deployment

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 deployment:**
- 1 On the Deployments tab, click the name of the deployment you want to modify.
 - 2 Make the changes you want to the deployment, 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.
- **To upload a new web application**
- 1 On the Deployments tab, click the name of the deployment you want to modify.
 - 2 Click on the green up arrow to the right of the web application you want to replace
 - 3 Browse to a the new .war file, click Open
 - 4 Click Add to upload the new .war file
 - 5 Click Save

- 6 The deployment is now updated, but will appear with a yellow icon (for unreconciled) until it is deployed



Viewing History on a Deployment

Each time you modify, deploy, or undeploy, a new version of the deployment is created. To see what has changed over time, select the deployment and click on the History tab.

Note Deployments can be restored from history using the Restore button.



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.

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:

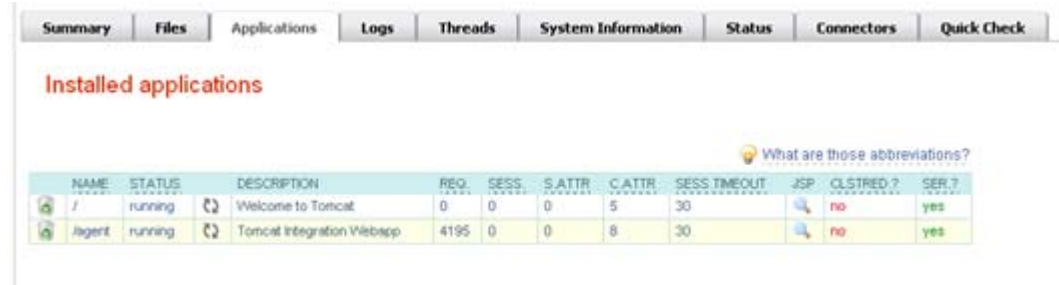
- 1 On the Files tab, click **New File**.
- 2 Enter the name you want to display for this file.
- 3 Specify the file (you can click **Browse** to select it from your file system) and click **Upload**.

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.



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.

Column	Description
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.

SummaryFilesApplicationsLogsThreadsSystem InformationStatusConnectorsQuick Check

Logs

show all

CLASS	FILE NAME	SIZE	MODIFIED	TYPE
jdk	C:\Tcatagent\Vogs\catalina.2009-12-14.log	4Kb	2009-12-14 10:15:27.004	org.apache.juli.FileHandler
jdk	C:\Tcatagent\Vogs\localhost.2009-12-14.log	0b	2009-12-14 10:11:10.426	org.apache.juli.FileHandler
log4j	C:\Tcatagent\Vogs\tcat-agent-app.log	26Kb	2009-12-14 21:46:32.114	org.apache.log4j.DailyRollingFileAppender
jdk	C:\Tcatagent\Vogs\tcat-agent-sys.2009-12-14.log	2Kb	2009-12-14 10:15:27.004	org.apache.juli.FileHandler
log4j	C:\Tcatagent\Vogs\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 19.

Summary	Files	Applications	Logs	Threads	System Information	Status	Connectors	Quick Check
Running threads								
What are those abbreviations?								
ID	NAME	EXEC POINT	STATE	IN NATIVE	SUSP.	VIC	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	83620	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	926	
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.



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 (catal i na. base and catal i na. 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

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

SummaryFilesApplicationsLogsThreadsSystem InformationStatusConnectorsQuick Check

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

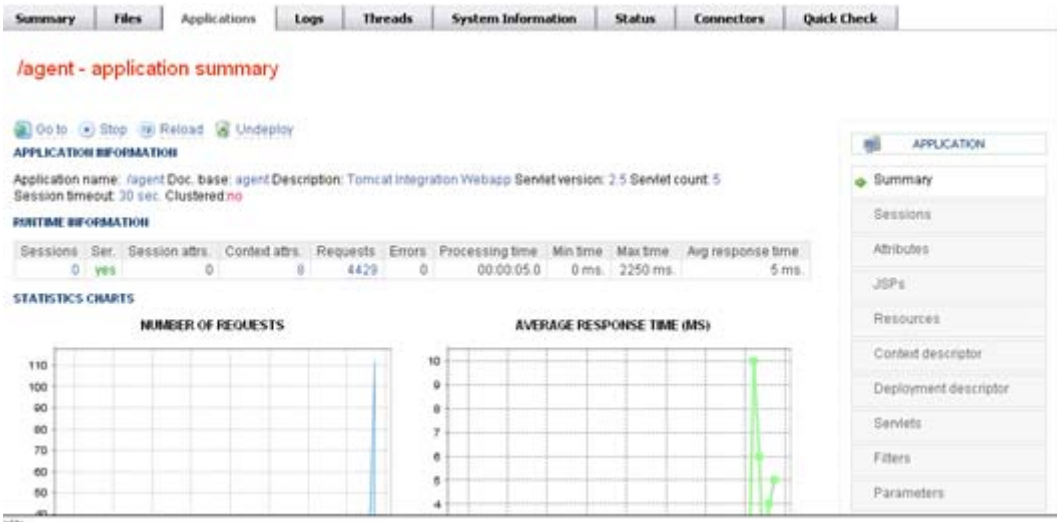
REMOTE IP	STAGE	PROC.TIME	IN	OUT	THREAD	URL
192.168.1.105	Service	00:00:00.0	375b	0b	http-51443-4	POST /agent/remoting/statusService
192.168.1.105	KeepAlive	00:00:06.0	0b	0b		
192.168.1.105	KeepAlive	00:00:01.0	0b	0b		
192.168.1.105	KeepAlive	00:00:06.0	0b	0b		

jk-8010

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

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.



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.

Working With Alerts

Alerts give you visibility to events that might need your attention and allow you to notify other administrators about actions they need to take. Alerts can be triggered on several different things:

- A server going down
- Web application statistics, such as average processing time
- Regular expressions in log files
- URL status check
- JMX attributes which match certain threshold criteria

When an alert occurs, a pop-up appears briefly and indicates that the alert condition occurred. You can also configure various alert destinations, such as an email address or an SNMP trap target.

Overview of Alerts

Click the Alerts tab to manage and view alerts. Use the navigation tree on the left to view raised alerts (that is, alerts that have been triggered), all alerts that have been set whether triggered or not, destinations, and notifications. Use the same nodes to manage alerts, destinations, and notifications. When you first click the Alerts tab, the pane displaying raised alerts appears.

Viewing Raised Alerts

Click the View raised alerts node in the navigation tree on the left to view alerts that have been triggered. The details pane displays a table with summary information about each alert, plus icons that indicate the severity of the alert. Alerts that have not yet been read appear bolded, as shown in the figure below. In addition, the number of unread alerts appears highlighted in red at the top of the pane.

MuleSoft™ | Tcat Server™
 Welcome, Administrator | 2 Unread Alerts | Help | Log Out

Dashboard Servers Deployments Alerts Repository Administration

Alerts View Alerts

Alerts let you see a summary list of SLA notifications and details about each SLA. Alerts not yet read are flagged for your attention.

Severity	Server	Name	Summary	Date
Critical	192.168.86.135	8181 Up	Server 192.168.86.135-8181 came online.	Tue Oct 05 17:54:40 GMT-700 2010
Critical	192.168.86.135	8380 down	Server 192.168.86.135-8380 went offline.	Tue Oct 05 17:45:31 GMT-700 2010

Page 1 of 1

Displaying 1 - 2 of

The details pane shows the following information about each alert:

- Severity of the alert. The severity of an alert is set when the alert is created, and can be one of the following: Fatal, Critical, Major, Minor, Information.
- Server on which the alert occurred
- Name of the alert
- Summary information about the alert, such as low memory alert
- Time stamp indicating the date and time the alert was raised

Click anywhere on an alert row to select that alert and mark that alert as read. You can also click the Mark as Read button to mark checked alerts as read, or click the Delete button to delete checked alerts. Once an alert is read, its font changes from bold to plain.

The icons to the left of each alert indicate the severity of the alert. Hover the mouse over these icons to see what they represent. Briefly, the icons indicate the following

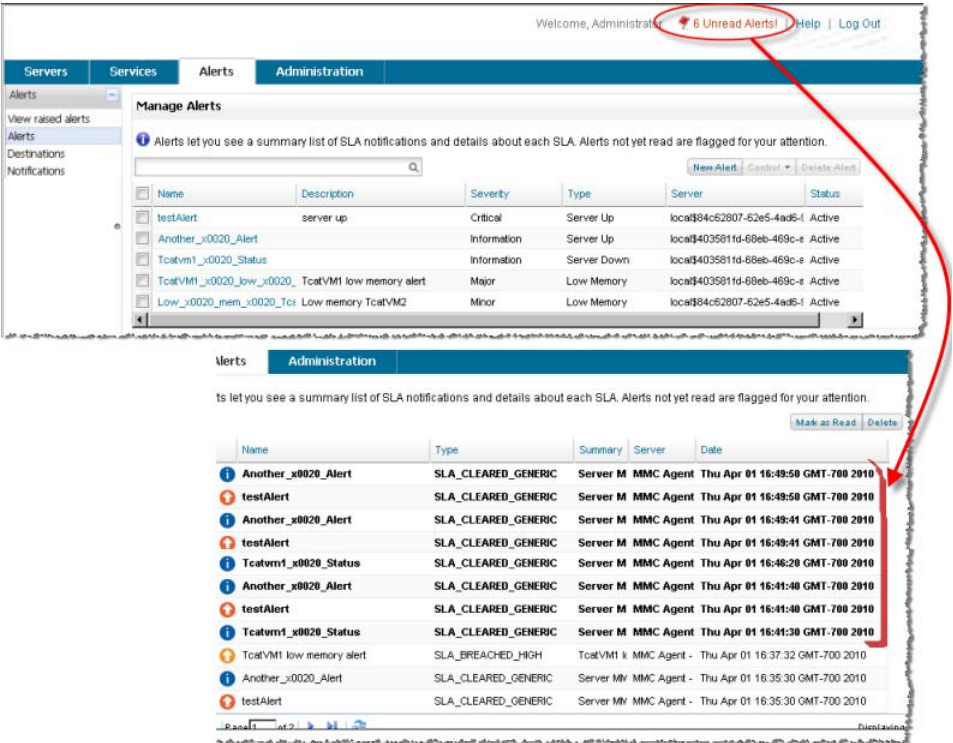
- The letter "i": Information alert
- Down arrow on blue background: Minor alert
- Up arrow on yellow background: Major alert
- Up arrow on orange background: Critical alert
- Tool icon on red background: Fatal alert

New Alerts Notification

The Alerts screen displays a message in red at the top notifying you of the number of alerts that have not yet been read. This notification about unread alerts appears at the top of all console panes. In addition, when an alert is triggered, a pop-up appears briefly indicating the alert that was triggered. You see this pop-up regardless of the pane you are currently viewing.

The new alert also increments the unread alerts counter, assuming you haven't yet looked at that alert. In addition to incrementing the counter, a note appears indicating the number of new alerts just added.

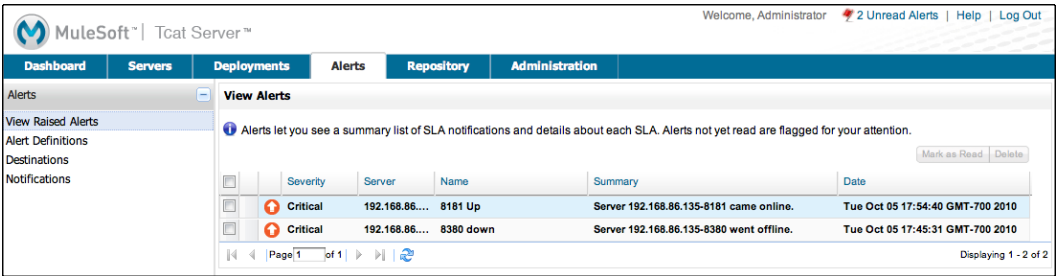
Click the notification of unread alerts, circled in red in the figure below, to open the pane to view alerts.



When the pane displaying raised alerts opens, notice that any unread alerts appear in bold font at the top of the pane. Alerts that have already been read are in plain font at the bottom of the pane, and the counter of unread alerts is decremented. The number of newly added alerts is also noted. Click an alert to read it.

Defining Alerts

Click the Alerts Definitions node in the navigation tree to see all alerts currently set up. From this pane, you can create new alerts and edit details about existing alerts. The figure below illustrates the details pane for viewing and managing alerts. The pane displays a summary of the existing alerts and provides buttons for adding new alerts, controlling alerts, and deleting them.

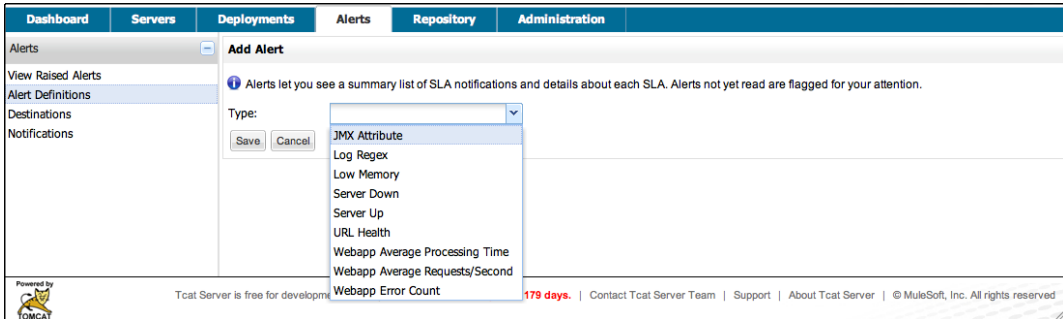


The details pane lists all currently set alerts and displays the following information about each alert:

- The name of the alert
- A description of the alert
- The alert severity level (Fatal, Critical, Major, Minor, Information)
- The type of the alert (such as Low Memory, Server Down, Server Up, Thread Pool SLA). You can also have alerts based on custom expressions.
- The identity of the server for which the alert applies, if any.
- The current alert status (Active, Disabled).

Creating New Alert

Click the New Alert button to create a new alert. The Add Alert pane displays a Type field. From the Type field's pull-down list, select the alert type.



Following are the names of the alert types and a description of the criteria they can be configure to fire on:

Alert Types

Type	Description
JMX Attribute	A JMX attribute exceeds, goes below, or matches a particular value
Log Regex	A line in a selected log file matches a particular regular expression
Low Memory	A JVM memory pool exceeds a threshold.
Server Down	A server goes offline or becomes unresponsive.
Server Up	A server comes online.
URL Health	Checks URLs periodically to ensure that they are available on the network.
Webapp Average Processing Time	Average request processing time for a web application.
Webapp Average Requests/second	Average requests per second for a web application.
Webapp Error Count	The number of errors in a one minute interval for a web application.

Note Currently, low memory alerts are not supported in an OpenJDK environment.

Once you select the alert type, the pane expands to display additional fields to which you enter data describing the alert. Different alert types have different data fields. The Add Alert pane displays the appropriate fields for the alert type you selected and you then enter the required and optional data. For example, the following figure shows the Add Alert pane for the Server Down alert type.

Dashboard

Servers

Deployments

Alerts

Repository

Administration

Alerts

View Raised Alerts

Alert Definitions

Destinations

Notifications

Add Alert

A Server Down Alert type allows you to detect when a server has been shutdown or restarted

Type:

Server Down

Name:

ProdServer Down

Description:

Severity:

Critical

Server:

192.168.86.135-8080

Active:

☒

Save

Cancel

Enter a name for the alert and, optionally, a description. From the Severity pull-down list, select the severity level (Fatal, Critical, Major, Minor, Information) of the alert. For the Server field, select the server(s) to which the alert applies (the default is none), and then check if the alert is active. A field for which data is required is marked with an exclamation point and highlighted in red if left blank. (The figure below has a field marked as required.)

The figure below shows a sample entry for a server down alert type. Once you select Server Down for the alert type, the screen expands and displays the additional fields of data you must enter. While Description is optional, you must select or enter data for the other fields. Click Save to save the new alert, or Cancel to quit without saving the alert.

Dashboard	Servers	Deployments	Alerts	Repository	Administration
Alerts					
Add Alert					
A Webapp Average Requests Alert type allows you to monitor an application for a specified average request threshold					
Type: Webapp Average Requests/Seco					
Name: pebble					
Description:					
Severity: Minor					
Server: 192.168.86.135-8380					
Active: <input checked="" type="checkbox"/>					
Application: All					
Comparison: All					
Value: /pebble (Pebble)					
/agent (Tomcat Integration Webapp)					
/pebbletest (Pebble)					
<input type="button" value="Save"/> <input type="button" value="Cancel"/>					

The following section explains the data you might select or enter for particular fields when setting up an alert.

Common Alert Fields

As explained previously, different alert types may prompt you for different data fields. Most alerts, regardless of type, include the following generic fields:

- Server—The server that this alert will be applied to.
- Severity—The severity of an alert is set when the alert is created, and can be one of the following: Fatal, Critical, Major, Minor, Information.
- Active—Whether or not the alert is currently active; if not, no alerts are triggered for this definition.

JMX Attribute

Configures an alert which fires if a JMX attribute value exceeds, goes below or equals a certain value.

- MBean and attribute—The MBean name and attribute value that is monitored. Click "Browse" to view the MBean tree and select an attribute.
- Comparison—> (greater than), < (less than), or = (equals to).
- Value—The value that will be used in the comparison to determine if this alert will fire.

- **Period**—How often in milliseconds the value is checked.

Log Regex

Configures an alert which monitors file changes for a particular regular expression.

- **Log Files**—A wildcard expression, such as "logs/*.log", which configures which log files to monitor. The directory is resolved relative to CATALINA_HOME.
- **Regular Expression**—The regular expression to look for in the log files. For instance, type "SEVERE" to find all SEVERE log warnings.

Low Memory

Configures an alert which monitors a specified memory pool to see if it exceeds a particular threshold percentage.

- **Memory Pool**—The memory pool to monitor, such as Total or PermGen.
- **Threshold Percentage**—The percentage of memory pool utilization that must be reached before firing the alert.

Server Down

Configures an alert which fires if a server goes down or becomes unresponsive. No extra configuration options.

Server Up

Configures an alert which fires if a server comes online. No extra configuration options.

URL Health

Configures an alert which monitors the status of a URL. Useful for monitoring whether a particular public URL is accessible.

- **URL**—The URL to check periodically.
- **Status Code**—The expected HTTP status code
- **Period**—The period, in seconds, with which to check the URL.

Webapp Average Processing Time

Configures an alert which monitors the average request processing time for a web application. This value is recomputed every minute.

- Application—The web application to check. Select "All" to monitor the request time across all applications.
- Comparison—> (greater than), < (less than), or = (equals to).
- Value—The value that will be used in the comparison to determine if this alert will fire.

Webapp Average Requests/second

Configures an alert which monitors the average requests per second for a web application. This value is recomputed every minute.

- Application—The web application to check. Select "All" to monitor the requests per second across all applications.
- Comparison—> (greater than), < (less than), or = (equals to).
- Value—The value that will be used in the comparison to determine if this alert will fire.

Webapp Error Count

Configures an alert which monitors the error count in the last minute for a web application. This value is recomputed every minute.

- Application—The web application to check. Select "All" to monitor the error count across all applications.
- Comparison—> (greater than), < (less than), or = (equals to).
- Value—The value that will be used in the comparison to determine if this alert will fire.

Editing an Alert

Click the alert name in the Manage Alerts summary pane. The pane displays the data previously entered for the alert, in the same format used for creating an alert, and you can make the desired changes.

Note that you cannot change the name of an alert. If you need to change the name of the alert, you should delete it and re-create it with the new name.

Managing Alerts

To manage alerts, click the alert to select it. A checkmark appears in the alert's checkbox when selected. Use the Control button to either activate or disable the selected alert(s). Use the Delete Alert button to delete the selected alert(s). Before alerts are actually deleted, a dialog prompts you to confirm the deletion.

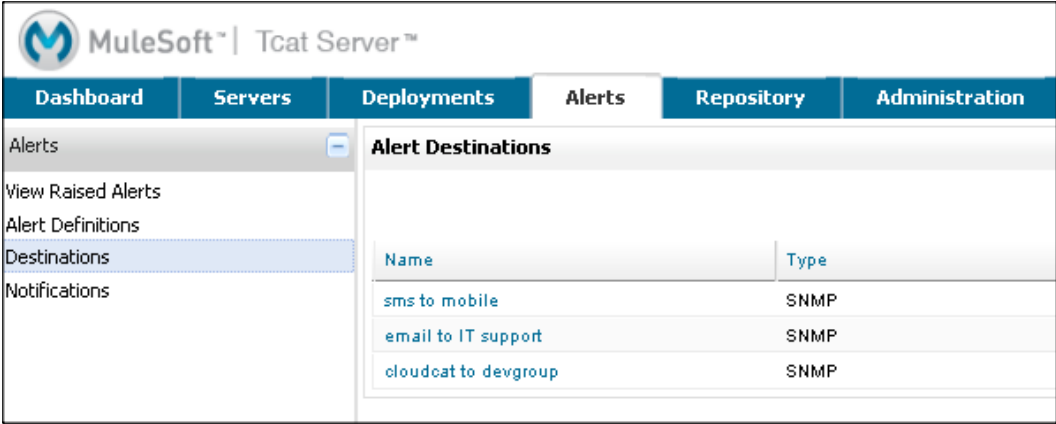
Defining Notifications and Destinations

Once you've defined your alerts, you can create email or SNMP destinations to send them to. Notifications configure which alerts are sent to which destination. For example, if you wanted to send all fatal alerts to ops@mulesoft.com, you could create an email destination with your SMTP server and then configure a notification to send fatal alerts to that email address using the SMTP server defined in the destination.

Setting Up Alert Destinations

You can set up alert destinations, which are the locations notified when alerts are triggered. You need to set up destinations prior to creating any notifications, since notifications are associated with specific destinations. Setting up a notification is described in “[Specifying Notifications](#)” on page 78.

Click the Destinations node in the navigation tree to see currently defined destinations. The pane displays the user-supplied name of the destination and the destination type, either email or SNMP.



The screenshot shows the MuleSoft Tcat Server interface. The top navigation bar includes tabs for Dashboard, Servers, Deployments, Alerts, Repository, and Administration. The Alerts tab is selected. On the left, a navigation tree lists Alerts, View Raised Alerts, Alert Definitions, Destinations (highlighted), and Notifications. The main content area is titled 'Alert Destinations' and displays a table with the following data:

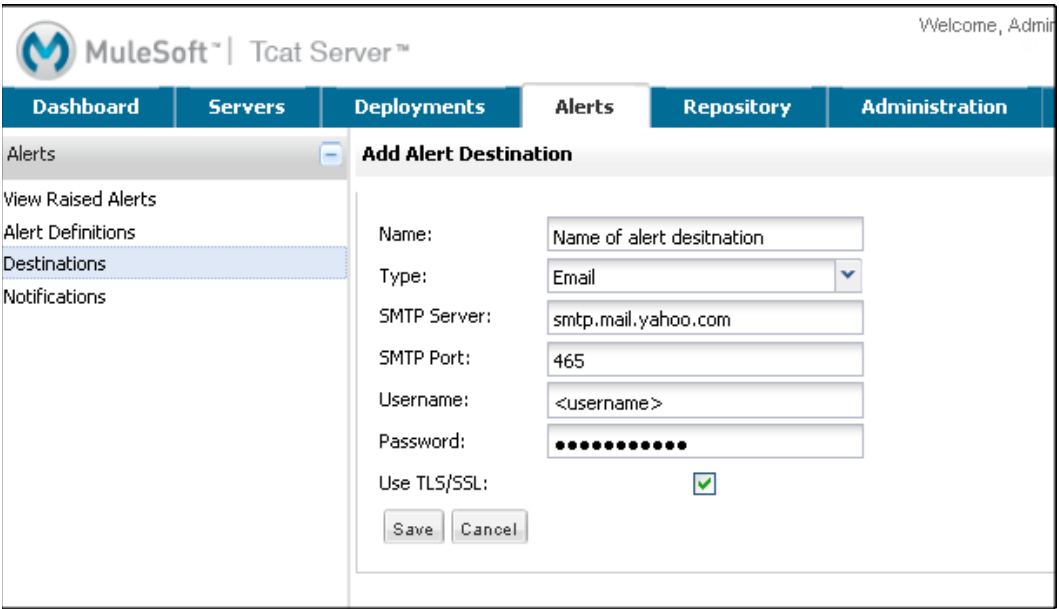
Name	Type
sms to mobile	SNMP
email to IT support	SNMP
cloudcat to devgroup	SNMP

Click the Add button to create a new destination. The Add Alert Destination pane opens. You start by entering a name for the destination and selecting the destination type. From the Type pull-down list, select the type of destination: email or SNMP.

Depending on the destination type you select, you are prompted for additional fields.

Configuring Email Destinations

When adding an email-type destination, you must designate server and the recipient user information. The figure below is an example of configuring an email destination:



For an email destination, you need to specify the following:

- SMTP server designation: You can identify the SMTP server by its IP address, such as 192.168.0.1, or by its url, such as smtp. mycompany. com.
- SMTP server port number: The server port number, such as 25.
- User email name: The email name and address for the recipient, such as user@mycompany. com.
- Password: The recipient's email password.

Check the TLS/SSL box if the server requires a secure environment. You must set this field to conform to your SMTP provider's server configuration.

When finished, click Save to complete adding the destination.

Examples of Email Destinations

The following examples show how you set up email destinations for Google and Yahoo mail recipients.

For Google mail (gmail), set the fields as follows:

Google Mail Settings

Field	Input Data
Name	Name of the alert destination
Type	Email
SMTP Server	smtp.googlemail.com
SMTP Port	465
Username	<username>@gmail.com
Password	<email password>
Use TLS/SSL	Yes

For Yahoo mail, set the fields as follows:

Yahoo Mail Settings

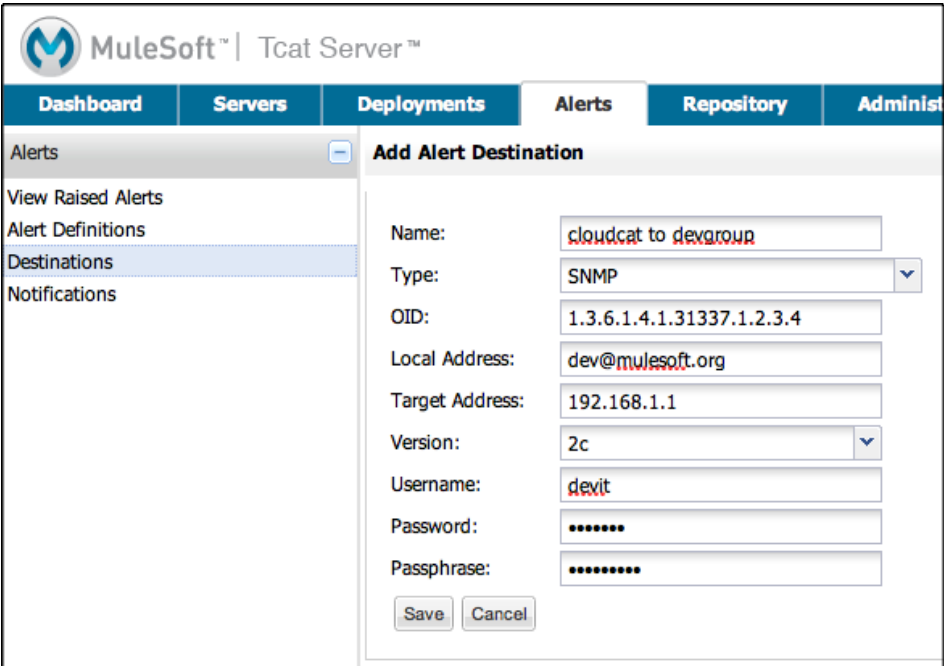
Field	Input Data
Name	Name of the alert destination
Type	Email
SMTP Server	smtp.mail.yahoo.com
SMTP Port	465

Yahoo Mail Settings

Field	Input Data
Username	<username>
Password	<email password>
Use TLS/SSL	Yes

Configuring SNMP Destinations

SNMP-type destinations have different fields than email-type destinations. The figure below is an example of how you might set up an SNMP destination:



Specify the following for SNMP destinations:

- **OID:** An identifying name for the destination. The OID you enter needs to conform to SNMP conventions. For example, a valid OID might be 1. 3. 6.

- **Local address:** The local SNMP address to which to bind the destination. The address must conform to SNMP conventions. An example of a local address might be `udp: 127. 0. 0. 1/40162`.
- **Target address:** The SNMP target address to which to send the alert notification. This address also must conform to SNMP conventions. For example, you might enter an address such as `udp: 127. 0. 0. 1/40163`.

When finished, click **Save** to complete adding the destination.

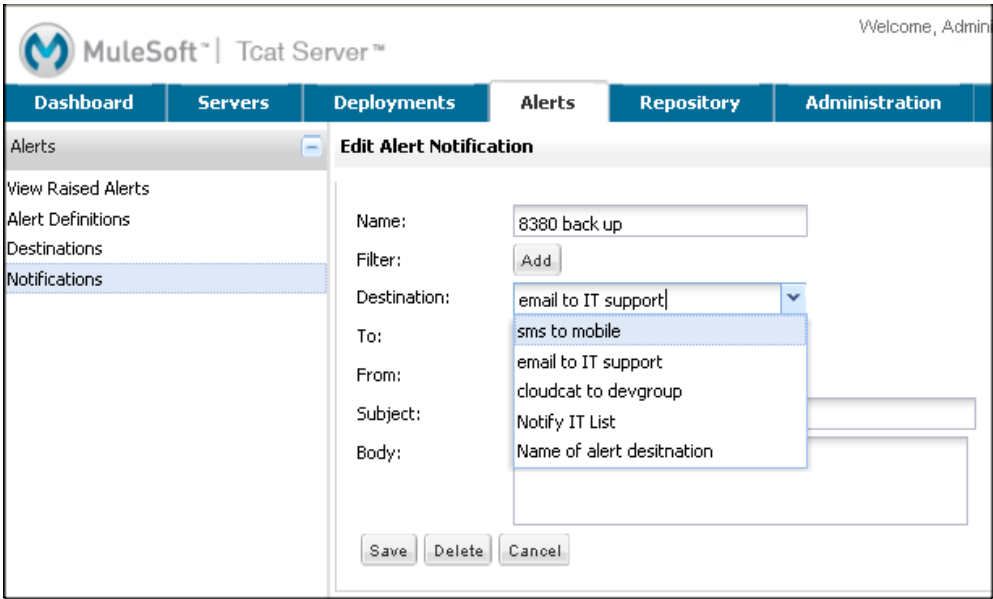
To edit a destination, double-click the destination entry in the **Alert Destinations** pane. The edit pane displays the data entered for that destination and you can change values as needed.

Specifying Notifications

Set up notifications after you have defined possible destinations for your notifications. When you create a new notification, you must specify a destination for the notification, and the destination must already exist.

Click the **Notifications** node in the navigation tree to view alert notifications that are already defined. The pane displays the names of all alert notifications and their destinations.

To create a new notification, click the **Add** button to open the **Add Alert Notification** pane. (See the figure below.)



Enter a name for the notification. You also need to specify one or more filters. When you click the Add button in the middle of the pane, the pane displays a table into which you specify the filter field(s) and values for these field. You may add as many filter fields as you like. Select each filter field from the Field pull-down menu. The Value pull-down menu displays the values that correspond to the selected field. (The figure below is an example of this.)

The screenshot shows the MuleSoft Tcat Server web interface. The top navigation bar includes tabs for Dashboard, Servers, Deployments, Alerts (selected), Repository, and Administration. On the left, a sidebar menu lists Alerts, View Raised Alerts, Alert Definitions, Destinations, and Notifications (highlighted). The main content area is titled 'Edit Alert Notification'. It contains the following fields and controls:

- Name:** Text input field with the value '8380 back up'.
- Filter:** A table with two columns: 'Field' and 'Value'.


Field	Value
Severity	Information

Below the table are 'Save' and 'Cancel' buttons, and an 'Add' button to the left.
- Destination:** A pull-down menu showing 'sms to mobile'.
- To:** Text input field with the value '222-333-4444@vtext.com'.
- From:** Text input field with the value 'MuleIT'.
- Subject:** Text input field with the value 'Server 8380 back up'.
- Body:** Text input field with the value 'please call #444-333-3434'.

At the bottom of the form are 'Save', 'Delete', and 'Cancel' buttons.

For each filter field you specify, be sure to click the Save button beneath the Field/Values table to save that filter, or click Cancel to discard the entry. You can select filter fields and delete them, if you want. Add additional filters by clicking the Add button, but be sure you click Save or Cancel before specifying these additional filter fields.

For each notification, you must select the destination from the Destination pull-down list. If you have not yet defined a destination, then this list is empty and you cannot complete the notification specification.

 MuleSoft™ | Tcat Server™

Dashboard

Servers

Deployments

Alerts

Repository

Administration

Alerts

View Raised Alerts

Alert Definitions

Destinations

Notifications

Edit Alert Notification

Name:8380 back up

Filter:

Field

Value

SeveritySeverity

InformationInformation

SaveCancel

Add

Destination:sms to mobile

To:sms to mobile

From:email to IT support

Subject:cloudcat to devgroup

Subject:Notify IT List

Body:please call #444-333-3434

SaveDeleteCancel

If the destination type you select is email, then the pane prompts you for the email recipient name, the name of the sender, a subject, and text for the body of the email. You must enter data for all these fields (See the figure below.)

The screenshot shows the MuleSoft Tcat Server web interface. The top navigation bar includes links for Dashboard, Servers, Deployments, Alerts (selected), Repository, and Administration. On the left sidebar, under the Alerts section, there are links for View Raised Alerts, Alert Definitions, Destinations, and Notifications (which is highlighted). The main content area is titled 'Edit Alert Notification'. It contains the following fields and controls:

- Name:** A text input field containing '8380 back up'.
- Filter:** A section with two dropdown menus. The first is labeled 'Field' and has 'Severity' selected. The second is labeled 'Value' and has 'Information' selected.
- Buttons:** Below the filter section are 'Save' and 'Cancel' buttons.
- Add:** A button located below the filter section.
- Destination:** A dropdown menu with 'email to IT support' selected.
- To:** A text input field containing 'ITList'.
- From:** A text input field containing 'IT@mulesoft.com'.
- Subject:** A text input field containing '8380 back up FYI'.
- Body:** A large text area containing 'Reset'.
- Bottom Buttons:** 'Save', 'Delete', and 'Cancel' buttons are located at the bottom of the form.

For email destinations, you may use an expression in the subject and body fields. To use an expression and have it correctly interpreted, you must enclose the expression in curly braces and precede it with a dollar sign (\$) symbol, as shown below. For example:

`${expression}`

You can combine the expression with some other text. For example, you might want a subject line that displays a standard message that an alert has been triggered but also includes an expression indicating the alert condition. You might set the subject field as follows:

Destination: emailTest

To: admin

From: user@gmail.com

Subject: You have an alert! \${description}

Body:

See “[Properties Used in Alert Expressions](#)” on page 84 for more information on properties that you can use with these alert expressions.

When the destination type is SNMP, you are prompted just for a value, as shown below. You must enter data for the Value field to be able to save the notification.

MuleSoft | Tcat Server™

Welcome, Administrator

Dashboard Servers Deployments Alerts Repository Administration

Alerts

View Raised Alerts

Alert Definitions

Destinations

Notifications

Add Alert Notification

Name: 8380 back up but dblcheck logs

Filter:

Field	Value
Server	192.168.86.135-8380

Add

Destination: Name of alert destination

To: it@muleservice.org

From: it@mule.org

Subject: \${serverName}.id

Body: Attention: at \${timestamp}, a \${severity} error occurred on the \${serverName}.

Save Cancel

For all notifications, be sure to click the Save button after you have entered all required data. Notice that the Save button is grayed out until all required fields are completed.

To edit a notification, double-click the notification entry in the Alert Notifications pane. The edit pane displays the data entered for that notification and you can change values as needed.

Properties Used in Alert Expressions

There are a number of properties whose values you can incorporate into alert destinations and notifications. You incorporate these properties using the expression syntax shown above. Some properties are common to all alert types, while others pertain to specific alert types.

The following properties are common to all alert types. (The class `com.mulesoft.console.alert.RaisedAlert` contains the declaration of alert properties.)

Alert Type Properties

Property	Description
id	Service identifier
name	Service name
serverId	Server identifier (currently not available in 2.2.6 release)
serverName	Server name (currently not available in 2.2.6 release)
description	Description of the raised alert
timestamp	Time when the alert was raised
source	Source of the raised alert
severity	Severity of the alert, such as Fatal or Critical
type	The type of the alert, such as Low Memory or Exception

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 54 and “[Monitoring Applications](#)” on page 62.

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-servlets.xml` file on that server and add or remove directories from the `writeExcludes` property in the `accessControlFileService` bean.

Chapter 4

Managing Cloudcat Instances

MuleSoft Cloudcat is enterprise Apache Tomcat in the cloud. Cloudcat is a fully supported virtual image (Amazon EC2 AMI) that allows you to deploy your Tomcat applications to the cloud. Cloudcat includes Apache Tomcat with a preconfigured MySQL database on Ubuntu Linux.

If you use MuleSoft Tcat Server, you can manage your Cloudcat instance(s) using the Tcat Server console. Cloudcat users get unlimited email-based support.

If you want to use GoGrid instead of Amazon EC2, see <http://www.mulesoft.org/documentation/display/TCAT/Using+Cloudcat+with+GoGrid>. If you want to upload your own version of Tomcat as an AMI to EC2 using the EC2 plug-in, see Chapter 5, “[Deploying to Amazon EC2](#)”.

- “[Prerequisites](#)” on page 87
- “[Creating a Cloudcat Instance](#)” on page 88
- “[Using Tcat Server to Manage Cloudcat Instances](#)” on page 89
- “[Deploying Tomcat Applications to Cloudcat](#)” on page 90
- “[Getting Support for Cloudcat](#)” on page 90

Prerequisites

Before you can use Cloudcat on Amazon EC2, you must meet the following prerequisites:

- Basic knowledge of Ubuntu commands
- Basic knowledge of Amazon EC2 and security groups
- Amazon EC2 account
- Download the EC2 tools (from <http://developer.amazonwebservices.com/connect/entry.jspa?externalID=351>) if you will open ports manually instead of through security groups

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

Purchase a Subscription to Cloudcat

If you have not yet purchased a subscription to Cloudcat, do the following:

- 1 Purchase the Cloudcat subscription here. If prompted, enter your credentials and payment information and complete the purchase.
- 2 On the registration page, enter your information and then click Register.

Creating a Cloudcat Instance

To use Cloudcat, you create a new instance based on the purchased Cloudcat AMI. You will enter one of the following AMI IDs depending on how much memory you want to use for the instance:

- 32-bit AMI: `ami-adf01cc4`
- 64-bit AMI: `ami-61f71b08`

For example, you can create the instance using the AWS Management Console as follows:

- 1 Launch the **AWS Management Console** to create one or more new instances.
- 2 In the console, click AMIs in the left-hand navigation pane, and then enter the AMI ID into the text box and press Enter.
- 3 Select the AMI and click Launch.
- 4 Follow the instructions in the wizard to create your instance(s). When asked to specify a security group, you can select an existing group (such as Default) or create a new security group just for Cloudcat that opens your HTTP and TCP ports (8080 and 51443 by default). You could then create all your Cloudcat instances with this security group and automatically open the required ports for all of them. Note that you can change the default HTTP and TCP ports for Tomcat on the instance by modifying the `server.xml` file and restarting Tomcat. In this case, you must open the new ports instead of the defaults.

After you complete the instructions in the wizard, the instance is created, which can take several minutes. Tomcat will then start in `/opt/tomcat` on the instance. You are now ready to deploy applications to Tomcat in your Cloudcat instance. If you want to use Tcat Server to manage your instance and to deploy your applications, see the next section.

Using Tcat Server to Manage Cloudcat Instances

Cloudcat includes the Tcat Server agent, so you can pair the Cloudcat instance with your Tcat Server instance and manage it via the Tcat Server console.

To enable access to Tcat Server, this will open up ports 8080 and 51433 by default.

- 1 SSH into your remote Cloudcat instance using the following syntax: `ssh -i <your private key> ubuntu@ec2-<public address>` where `<your private key>` is the private key you set up for your EC2 account and `<public address>` is the address for the instance. In the Amazon console, the public address is displayed in the bottom pane when you click the instance.
- 2 If your remote Tomcat server instance is stopped (it started automatically when you created the instance), start it now by entering the following command: `sudo /etc/init.d/tomcat start`
- 3 If you did not open your HTTP and TCP ports via the security group, you must open them manually on the remote instance by entering the following command:
`$ ec2-authorize -p <port> <securityGroup>` where `<securityGroup>` is the name of the group you specified when you created the instance.

To register the remote Tomcat:

- 1 Start your local Tcat Server instance and console.
- 2 On the Servers tab in the console, click New Server.
- 3 Give the server a unique name, enter the URL `http://<public address>:8080/agent` (replace `<public address>` with the instance's public address), and then click Register.

You are now ready to manage the Cloudcat instance from your local Tcat Server console. For more information, see “[Monitoring a Server](#)” on page 54.

Deploying Tomcat Applications to Cloudcat

After you have set up Cloudcat, you can start deploying your Tomcat applications in the cloud. You can manually deploy your current applications to Cloudcat by simply copying the application WAR files from your local Tomcat server and pasting them into the webapps directory of the Cloudcat instance on EC2 or GoGrid, or you can use the Tcat Server console to perform the deployments and configuration. For more information on using Tcat Server for deployment, see [“Deploying Applications”](#) on page 50.

Note that your web application may need specific configuration including configuring of MySQL drivers. You can use server profiles in Tcat Server to make configuration changes on Cloudcat instances. For details, see [“Working with Server Profiles”](#) on page 118.

After you have deployed a web application to your Cloudcat instance, restart Cloudcat either manually or through Tcat Server console, and then test your application in the cloud.

Getting Support for Cloudcat

Your subscription includes unlimited support via email. To request support, send your query to tcats-install@mulesoft.com, and a support representative will contact you.

Chapter 5

Deploying to Amazon EC2

The EC2 plugin allows you to create Amazon machine instances (AMIs) of your existing Tomcat instances and deploy them to EC2 via the Tcat Server console. This chapter describes how to install the plug-in, create the AMI, and deploy it. It includes the following sections:

- “**Installing the Plug-in**” on page 91
- “**Creating an Amazon EC2 Account**” on page 92
- “**Opening Ports**” on page 92
- “**Amazon Machine Images**” on page 92
- “**Using the Plug-in**” on page 93
- “**Creating a Tcat Server AMI**” on page 94
- “**Configuring Tcat Server for Automatic Startup**” on page 95

Note If you want to use an existing, fully configured cloud implementation of Tomcat instead of creating and deploying your own instance, see “Using Cloudcat with Amazon EC2” at

<http://www.mulesoft.org/documentation/display/TCAT/Using+Cloudcat+with+Amazon+EC2>

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 plug-in 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.

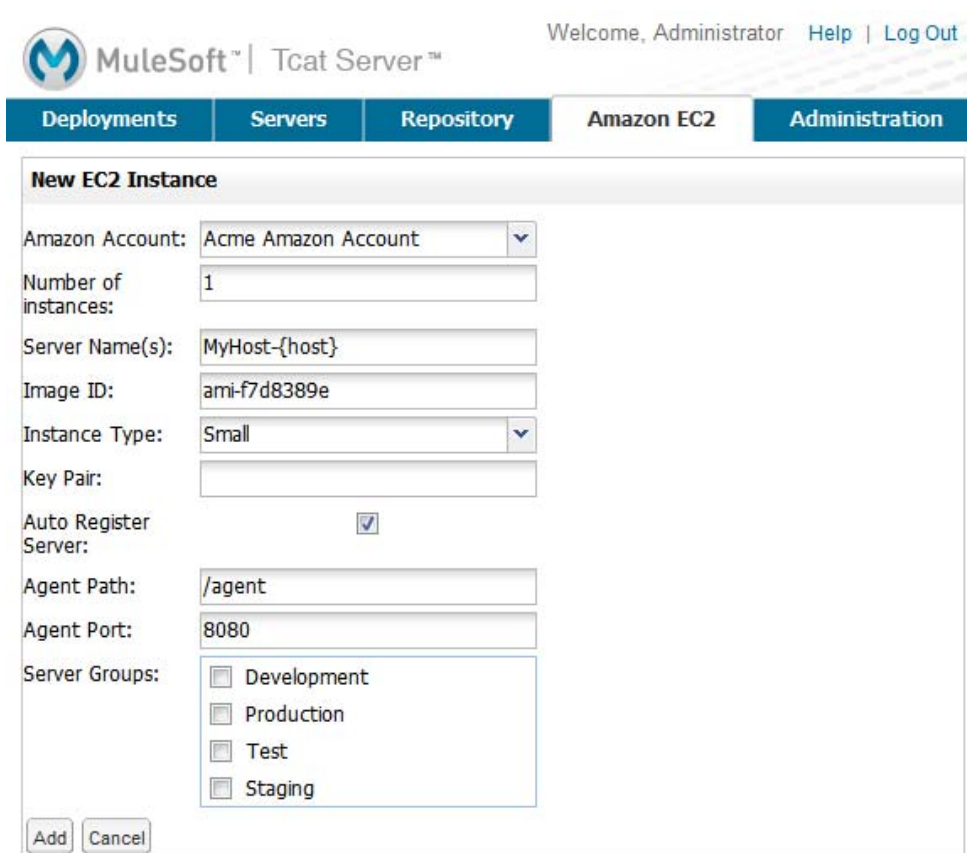
You will want to create your own AMI (see [page 94](#)) with a Tcat Server instance on it. For testing purposes, we provide an image with Tcat Server and Ubuntu 9.04 from Alestic (<http://alestic.com/>). On startup, this image will start Tcat Server and make it available on port 8080. Please note that 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**.



The screenshot shows the Tcat Server Administration console. At the top, there is a MuleSoft logo and the text 'Tcat Server™'. To the right, it says 'Welcome, Administrator' with links for 'Help' and 'Log Out'. Below this is a navigation bar with tabs: 'Deployments', 'Servers', 'Repository', 'Amazon EC2' (which is selected), and 'Administration'. The main content area is titled 'New EC2 Instance' and contains the following form fields:

- Amazon Account: A dropdown menu showing 'Acme Amazon Account'.
- Number of instances: A text input field with the value '1'.
- Server Name(s): A text input field with the value 'MyHost-{host}'.
- Image ID: A text input field with the value 'ami-f7d8389e'.
- Instance Type: A dropdown menu showing 'Small'.
- Key Pair: An empty text input field.
- Auto Register Server: A checkbox that is checked.
- Agent Path: A text input field with the value '/agent'.
- Agent Port: A text input field with the value '8080'.
- Server Groups: A list of checkboxes for 'Development', 'Production', 'Test', and 'Staging', all of which are currently unchecked.

At the bottom left of the form, there are two buttons: 'Add' and 'Cancel'.

► **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 92).
- 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.

Chapter 6

Working with the Repository

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

- [“Viewing the Repository Contents”](#) on page 98
- [“Modifying an Artifact or Workspace”](#) on page 98
- [“Adding a Workspace”](#) on page 99
- [“Adding Applications and Other Artifacts”](#) on page 99
- [“Subscribing to Change Notifications”](#) on page 100
- [“Setting Policies”](#) on page 64
- [“Setting Security”](#) on page 100
- [“Removing an Artifact or Workspace”](#) on page 101

Viewing the Repository Contents

The Repository tab allows you to add and remove *artifacts*, (such as configuration files and applications, and manage *workspaces*. A workspace is a location in the repository where artifacts are stored. You can upload all your artifacts into the default Applications workspace, or you can create new workspaces to organize your artifacts.

When you first view the Repository tab, it displays the top-level workspaces, including the default workspace Applications. Click a workspace to see the artifacts and child workspaces it contains. You can use the navigation tree on the left to navigate through the workspace levels and select the workspace you want to work with.

Modifying an Artifact or Workspace

When you click an artifact or workspace, its details appear in on four tabs:

- The Items tab displays any artifacts within this artifact, such as applications within a workspace or versions within an application.

- The Info tab displays the name, type, metadata (such as the default version), and comments for this artifact. Adding metadata allows you to specify values for properties that will be useful for searching artifacts in a future release. Adding comments allows you to enter freeform text that is for your own reference when viewing the object.
- The Security tab allows you to control access to this artifact (see “[Setting Security](#)” on page 100)

Adding a Workspace

You add a workspace when you want to create logical groupings of your artifacts. This is useful as the number of artifacts in your repository grows. You can create workspaces within workspaces to further organize your applications.

► **To add a workspace:**

- 1 On the Repository tab, navigate to the workspace in which you want to add the new workspace, or just stay in the default view to add it to the root level.
- 2 Click **New Workspace**.
- 3 Enter a unique name for this workspace and click **Add**. Workspace names should be meaningful as well as unique. For example, when creating server profiles, you might want to create a parent workspace called `ServerProfiles` and then create workspaces under it called `Production`, `LinuxDevelopment`, and so on.

The workspace now appears in the navigation tree on the left, and the view changes to show the contents of the new workspace. You are now ready to add artifacts to this workspace.

Adding Applications and Other Artifacts

You can add artifacts to the repository such as applications, configuration files, or any other type of file that you want to deploy to multiple Tomcat servers. The following steps describe how to add an artifact manually to the repository. You can also add an artifact to the repository when you are [creating a package|Deploying Applications], or you can use the [Maven Publishing Plug-in] to cause applications to be uploaded to the repository every time they are built (see Appendix A, “[Using the Maven Publishing Plug-in](#)”).

► **To add an artifact:**

- 1 On the Repository tab, navigate to the workspace where you want to add the artifact.
- 2 Click **New Artifact**.
- 3 Browse to the location of this file and select it, or just type its full path name in the File box.
- 4 In the Name field, enter a unique name for this file. You can use the same name as the file name as long as it is unique within this workspace.
- 5 In the Version field, leave 1 set if this is the first time you are uploading this file, or enter a higher number if there is already a previous version of this file. Having multiple versions of a file makes it easy to roll back to a previous version if needed.
- 6 Click **Add**.

Subscribing to Change Notifications

Tcat Server provides an RSS feed of changes to its repository. You can subscribe to this feed to be notified about changes to a specific artifact or all the artifacts in a workspace. To subscribe, click the artifact or workspace, click **Feed**, and then specify the options provided by your browser for subscribing to this feed. If you just want to subscribe to comments changes, click the Atom feed icon next to Comments.

Setting Security

In Tcat Server, users are assigned to roles, and those roles are granted specific privileges for reading, editing, and deleting artifacts. You can override these settings for a specific artifact or all artifacts in a workspace.

► **To set security on an artifact or workspace:**

- 1 Click a workspace or artifact on the Repository tab.
- 2 Click the Security sub-tab.
- 3 In the row for the role whose permissions you want to change, select whether you want to grant or revoke the right to read, modify, or delete the artifact(s), or to manage policies for the artifact(s).

- 4 Click Save.

Removing an Artifact or Workspace

Warning When you remove a workspace, all of its artifacts are also removed.

To remove an artifact or workspace, select it on the Repository tab, and then click **Delete**.

Chapter 7

Extending Tcat Server

This chapter describes how to create and run extension on Tcat Server. It includes the following sections:

- “**Automating Tasks**” on page 102
- “**Scripting Examples**” on page 106
- “**Using the REST API**” on page 110

Automating Tasks

The Admin shell allows you to easily build and install your own extensions for Tcat Server by writing scripts. You can then run the scripts on startup or schedule them using a cron-like scheduling mechanism. The scripts are written using the Java-like Groovy scripting language. For more information about Groovy, see <http://groovy.codehaus.org/>.

By default, Tcat Server includes two scheduled scripts:

- Heartbeat: pings all your servers looking for unregistered Tcat Server instances. You should not need to modify this script.
- Usage Statistics: sends anonymous information to MuleSoft. For more information, see “**Scheduling the Usage Statistics Script**” on page 121.

For more examples of scripts, see “**Scripting Examples**” on page 106. For information on controlling logging of scripts, see “**Shell Script Log**” on page 123.

Creating a Script

To create a script, go to the Administration tab, click **Admin Shell**, and then enter your script. For example, you could enter the following text:

```
println "Hello"
```

You can then click **Evaluate** to run the script once. If it works without any errors, you can save the script so you can schedule it to run as a scheduled job. To save the script, enter a unique name for it, and then click **Save**. If you always want this script to run when Tcat Server starts, click **Run on startup**.

To use an existing script as a template for a new script, click the existing script on the right side of the screen, make your modifications, and then click **Save As** and enter a new name for the script.

Modifying a Script

To edit an existing script, click its name on the right side of the screen, make the modifications, and then click **Save**. At any time, you can click **Reset** to return to the last saved version of the script.

To delete a script, click it on the right side of the screen, and then click **Delete**.

Scheduling a Script

You can use the scheduler to periodically run your script, which is useful for jobs such as replication.

► To create a scheduled job:

- 1 On the Administration tab, click **Scheduler**, and then click **New**.
- 2 Select the script to run.
- 3 Enter a unique name and description for this scheduled job.
- 4 Enter a cron command to specify when the script should be run.
- 5 To allow this script to be run even if the last execution of the script is still running, click **Allow Concurrent Execution**.
- 6 Click **Save**.

Note After you click **Save** to save the scheduled job, the script will run on the specified schedule until you delete the job.

► To delete a scheduled job:

- 1 Click **Scheduler** on the Administration tab.
- 2 Click the scheduled job you want to delete.
- 3 Click **Delete**, and then click **OK** to confirm you want to delete this job.

Cron Command Syntax

You use the following syntax when specifying the cron command:

Field Name	Mandatory	Allowed Values	Allowed Special Characters
Seconds	YES	0-59	, - * /
Minutes	YES	0-59	, - * /
Hours YES	0-23	, - * /	
Day Of Month	YES	0-31	, - * / L W
Month YES	1-12 or JAN-DEC	, - * /	
Day Of Week	YES	1-7 or SUN-SAT	, - * / L #
Year	NO	empty, 1970-2099	, - * /

Following is a description of the special characters:

Character	Description
,	Separates individual values, such as 0,30 in the minutes position to run the job on the hour and half hour.
-	Specifies a range of values, such as MON-FRI in the Day Of Week position to run the job each day of the work week.
*	Specifies all values for that position, such as every day of the week when * is specified in the Day Of Week position.
?	Skips setting a value for that position. Since Day of Week and Day of Month are mutually exclusive, always use ? in one of these fields and specify * or a specific value for the other.

Character	Description
/	Specifies increments, such as every fifteen seconds starting on the first second of each minute when 1/15 is specified in the Seconds position.
L	Specifies the last day of the month or week, depending on the position.
W	Specifies the weekday nearest the specified day, such as running the job on the weekday closest to the third of the month when 3W is specified in the Day of Month position.
#	Specifies a day of the week as it occurs in the month, such as running the job the second Friday of every month by specifying 6#2 in the Day of Week position (where 6 is the sixth day of the week, or Friday, and #2 specifies the second occurrence of that day in the month).

Following are some examples of cron commands:

Command	Description
0 0 12 * * ?	12pm (noon) every day. Note that no year is specified, because the year position is optional.
0 15 10 ? * *	Any of these commands runs the job at 10:15am every day
0 15 10 * * ?	
0 15 10 * * ? *	
0 15 10 * * ? ?	
0 0/5 14 * * ?	Every 5 minutes starting at 2pm and ending at 2:55pm, every day
0 0/5 14, 18 * * ?	Fire every 5 minutes starting at 2pm and ending at 2:55pm, AND fire every 5 minutes starting at 6pm and ending at 6:55pm, every day
0 0/5 14-16 * * ?	Every 5 minutes starting at 2pm and ending at 4:55pm, every day
0 10, 44 14 ? 3 WED	2:10pm and 2:44pm every Wednesday in March
0 15 10 ? * 6L 2010-2012	10:15am on the last Friday of every month during the years 2010, 2011, and 2012

For more information on the cron command and its options, see

[http://www.opensymphony.com/quartz/wikidocs/CronTriggers%20Tutorial.htm](http://www.opensymphony.com/quartz/wikidocs/CronTriggers%20Tutorial.html)

|

Scripting Examples

This section provides examples of using scripts.

Redeploying a Named Package

This simple script grabs a package with a particular name and redeploys it using the provisioning service.

```
def provisioningService = applicationContext.getBean("provisioningService")
def deployment = provisioningService.getDeploymentByName("MyPackage")
provisioningService.redeploy(deployment.id)
```

Automatically Redeploying After an Artifact Update

If you're using the Maven plug-in or the API to update a WAR, you may want to listen for the update and automatically redeploy any packages associated with it. The following script installs an event listener, which listens for updates to an item's artifact property, which changes when the WAR file data has been changed. When this event occurs, the script searches for any deployment packages with this particular artifact and redeploys them.

Note In Tcat 6.2.0 there is a bug which prevents artifact updates from the REST API from being saved. To fix this issue, download this file:

<http://www.mulesoft.org/download/attachments/25854031/ItemCollection.class> and copy it into TCAT_HOME/webapps/console/WEB-INF/org/mule/galaxy/atom.

```
import com.mulesoft.tcat.DeploymentService;
import com.mulesoft.tcat.DeploymentAction.Action;
import org.mule.galaxy.Item;
import org.mule.galaxy.Registry;
import org.mule.galaxy.event.EventManager;
import org.mule.galaxy.event.PropertyChangeEvent;
import org.mule.galaxy.event.annotation.Async;
import org.mule.galaxy.event.annotation.BindToEvent;
import org.mule.galaxy.event.annotation.OnEvent;
```

```

import org.springframework.context.ApplicationContext;

@BindableEvent("PropertyChanged")
public class AutomaticRedeployer {
    def Registry registry;
    def DeploymentService deploymentService;
    def EventManager eventManager;
    def ApplicationContext applicationContext;

    public AutomaticRedeployer(ApplicationContext applicationContext) {
        this.applicationContext = applicationContext;
    }

    public void install() {
        registry = applicationContext.getBean("registry");
        eventManager = applicationContext.getBean("eventManager");
        deploymentService = applicationContext.getBean("deploymentService");

        // remove the previous listener, if any
        eventManager.listeners.each {
            if (it.className == "AutomaticRedeployer") {
                eventManager.removeListener(it)
                println "Removed previous listener"
            }
        }
        // add the event listener again
        eventManager.addListener(this)
    }

    @Async
    @OnEvent
    void onEvent(PropertyChangeEvent e) {
        if ("artifact" != e.propertyName)
            return;

        println "Artifact for ${e.itemPath} has been updated. Looking for
        deployments..."

        doRedeploy(e.itemPath)
    }

    /**
     * Redeploy any packages which have the specified artifact version.
     */
    void doRedeploy(String artifactVersionPath) {

```

```

def query = "select where type.name = 'Deployment Version' and
package.latest = 'true' and deployment.artifacts = '${artifactVersionPath}' "

println "Query: ${query}"

def results = registry.search(query, 0, -1)

for (item deployment : results.results) {
    println "Redeploying ${deployment.parent.name}"

    deploymentService.doAction(Action.REDEPLOY,
        deployment.parent.id)
    }
}

new AutomaticRedeployer(applicationContext).install();

```

Sending Notifications of New Artifact Versions

This script watches for new artifact versions and sends an email to any user who is registered in the contacts property.

```

import org.mule.galaxy.event.*;
import org.mule.galaxy.event.annotation.*;

import org.mule.galaxy.Registry
import org.mule.galaxy.security.User
import org.mule.galaxy.Item

import javax.mail.*;
import javax.mail.internet.*;

import java.util.Properties;

// An event listener which fires emails when new entry versions are created
@BindToEvents(["EntryVersionCreated"])
public class EmailNotifier {

    def String userProperty = "contacts"
    def String server = 'smtp.foo.com'
    def String port = '465'

```

```

def String username = 'XXXX'
def String password = 'XXXX'
def Registry registry

@Async
@OnEvent
void onEvent(EntryVersionCreatedEvent e) {
    notifyUsers(e)
}

void notifyUsers(ItemEvent e) {
    Properties props = new Properties();

    props.setProperty("mail.host", server);
    props.setProperty("mail.user", username);
    props.setProperty("mail.smtp.port", port);
    props.setProperty("mail.password", password);
    props.setProperty("mail.smtp.auth", "true");

    Item item = registry.getItemById(e.getItemId());

    def mailSession = Session.getDefaultInstance(props, null);
    Transport transport = mailSession.getTransport("smtps");

    MimeMessage message = new MimeMessage(mailSession);
    message.setSubject("Artifact/entry ${item.name} was created");
    message.setContent("Artifact/entry was created in ${item.path}",
        "text/plain");

    def contacts = item.getProperty(userProperty);
    contacts?.each {
        message.addRecipient(Message.RecipientType.TO,
            new InternetAddress(it.email));
    }

    if (!contacts) { return };

    transport.connect();
    transport.sendMessage(message, message.getAllRecipients());
    transport.close();
}

// remove the previous listener, if any
eventManager.listeners.each {
    if (it.class.simpleName == "EmailNotifier") {

```

```

        eventManager.removeListener(it);
        println("Removed listener");
    }
}

// add the event listener again
eventManager.addListener(new EmailNotifier(registry: registry));

```

Replicating Workspaces

You can replicate workspaces from one Tcat Server instance to another. Replications are performed via custom scripts. This allows you to easily customize the replication process and schedule it via the Scheduler.

Replication occurs by copying artifacts and entries from one workspace to another. If you're replicating to a remote Tcat Server instance, you must first attach it as a remote workspace.

Once you have both your origin and destination workspace in mind, you can create a script that performs this replication. Following is an example script:

```

import org.mule.galaxy.ee.util.*;
// Create a replicator called "myReplicator"
def replicator = new Replicator(registry, activityManager, "myReplicator");

// Copy from the first workspace, to the second one.
replicator.copy("/LocalWorkspace", "/RemoteWorkspace");

```

This script copies all artifacts and entries from LocalWorkspace to RemoteWorkspace. Note that it is beneficial to give your replicators an ID, in this case `myReplicator`. This makes it easy to view the actions of the replicator in the activity log.

Once you have saved this script, it is easy to schedule this script to be run periodically via the scheduler (see [“Scheduling a Script”](#) on page 103).

Using the REST API

The REST API for Tcat Server provides a way for you to programmatically access much of Tcat's functionality. It allows you to:

- Register new servers
- Manage existing servers

- Manage server groups
- Restart servers
- Access and manage files on servers
- Create and manage packages
- Create, update, and delete workspaces and artifacts in the repository

For complete information on using the REST API, see
<http://www.mulesoft.org/display/TCAT/Using+the+REST+API>.

Chapter 8

Administering Tcat Server

The Administration tab allows users with the Administrator role to set up security, work with users and roles, and more. This chapter describes how to administer Tcat Server. It includes the following sections:

- **“Managing Users and Groups”** on page 112
- **“Enabling Authentication Through LDAP”** on page 115
- **“Managing Lifecycles”** on page 86
- **“Managing Policies”** on page 87
- **“Working with Server Profiles”** on page 118
- **“Creating Remote Workspaces”** on page 120
- **“Monitoring Repository Activity”** on page 120
- **“Applying Your Tcat Server License”** on page 121

Managing Users and Groups

Tcat Server uses the common permissioning scheme of users and roles. Users are individuals who can log in to Tcat Server. Roles allow you to assign permissions collectively to a group of users, such as granting certain users the Administrator role, which has permissions like Manage Users and Manage Properties assigned to it.

You manage users and roles on the Administration tab in Tcat Server. You can also use LDAP to enable authentication (see **“Enabling Authentication Through LDAP”** on page 115). Note that after you enable authentication through LDAP, the Users link on the Administration tab is no longer available.

Managing Users

This section describes how to create, edit, and delete a user in Tcat Server.

‣ **To create a user:**

- 1 On the Administration tab, click **Users**, and then click **New**.
- 2 Enter the user name that the user will enter when they log in to Tcat Server, as well as the user's full name, email address, and password (at least five characters).
- 3 Select at least one user group that you want this user to have.

The default user groups are Administrator and User. If you need to create more user groups, see “**Managing User Groups**” on page 114.

- 4 Click **Save**.

‣ **To edit a user:**

- 1 On the Administration tab, click **Users**, and then click the user name of the user you want to edit.
- 2 Edit the user's name or email address if needed.
- 3 If you need to reset the user's password, click **Reset Password**, enter the new password twice, and then click **OK**.
- 4 To assign a user group to the user, select it in the Available User Groups list and click **>**. To remove a user group from the user, select it in the Joined User Groups list and click **<**.
- 5 When you have finished editing this user, click **Save**.

‣ **To delete a user:**

- 1 On the Administration tab, click **Users**, and then click the user name of the user you want to delete.
- 2 Click **Delete**.
- 3 Click **OK** to confirm that you want to delete this user.

User Groups

Tcat comes with several pre-defined user groups with the following levels of permissions:

Pre-Defined User Group Permission Levels

User Group	Permission Level
Administrators	Users who have full control over a Tcat instance.
Deployers	Users with the ability to view deployments, deploy, undeploy and redeploy them.
Monitors	Users which have a read only view into the console.
Server Administrators	Users who can manage servers and deployments, but not users, groups or anything else on the administration tab.

Managing User Groups

This section describes how to add, edit, and delete a user group in Tcat Server. Note that you cannot delete the Administrator user group.

There are two categories of permissions in Tcat - those which are global (displayed on the left) and those which can be applied to server groups (displayed on the right). For instance, the ability to view servers is a server group permission. A user can have that permission for all server groups or a particular server group.

Let's say you want to give a user the ability to only view servers in the all server groups. To do this, select the group All from the drop down box and select the "Server - View" permission. If you wanted this user to only have the ability to view servers for the Production group, then you would uncheck the permission for the All group, select Production from the server group drop down box, and then select the "Server - View" permission.

To create a user group:

- 1 On the Administration tab, click User Groups in the left pane, and then click New.
- 2 Enter a name for this user group and click Save.

The new user group appears in the user group list.

- 3 Check the permissions you want users with this user group to have.
- 4 When you have finished creating the user groups you want, click Save.

To assign this user group to a user, edit the user as described above.

To edit a user group:

- 1 On the Administration tab, click User Groups.
- 2 Edit the assigned permissions as needed.
- 3 If you want to change the user group name, click its name, type the new name, and then click Save.

Note You cannot edit the Administrator user group.

- 4 When you have finished editing user groups, click Save.

To delete a user group:

- 1 On the Administration tab, click User Groups, and then click the name of the user group you want to delete.

Note You cannot delete the Administrator user group.

- 2 Click Delete.
- 3 Click OK to confirm that you want to delete this user group.

Enabling Authentication Through LDAP

Tcat Server can use an LDAP server to manage users and roles. To configure Tcat Server to use LDAP, you take the following steps:

- 1 Set up Tcat Server Roles (see below)
- 2 Add the LDAP JAR to your classpath (see [page 116](#))
- 3 Create the LDAP configuration file (see [page 116](#))

The rest of this section describes these steps in detail.

Set up Tcat Server Roles

If a user has a role (also called a *group*) on the LDAP server, and you want to use that same role to manage permissions in Tcat Server, you must also create that role inside Tcat Server and set the appropriate permissions. For instance, if you add the `Devel opers` role to the LDAP server, you must also create a role with that name in the Tcat Server Administration panel and configure the appropriate permissions.

This can be done in one of two ways:

- 1 Create a role called Administrator on the LDAP server. Tcat Server comes with a set of default permissions for users with the Administrator role. Ensure that your user name has the Administrator role, and then log into Tcat Server. You can then create other roles, assign the appropriate permissions, and remove the original Administrators role from the LDAP server.
- 2 Start a Tcat Server instance without LDAP enabled. Create roles in the Tcat Server instance that correspond to your roles on the LDAP server, and then follow the instructions below to enable LDAP.

Add the LDAP JAR to Your Classpath

The Tcat Server LDAP support JAR file contains the necessary classes for Tcat Server to authenticate against an LDAP server. You can download this file from <http://repository.muleforge.org/org/mule/galaxy/galaxy-ldap/2.1.1/galaxy-ldap-2.1.1.jar>. Download the file to the `/WEB-INF/lib` directory under the Tcat Server console web application (`TCatHome/webapps/console/WEB-INF/lib`).

Create the LDAP Configuration File

You must create an LDAP configuration file called `ldap.xml` and place it in `/WEB-INF/classes`. Following is an example template for this file, followed by a description of its properties:

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-2.5.xsd">

  <bean id="initialDirContextFactory"
```

```

        class="org.acegi.security ldap.DefaultInitialDirectoryContextFactory">
        <constructor-arg value="//host:port"/>
        <property
name="managerDn"><value>cn=Manager, dc=acme, dc=org</value></property>
        <property name="managerPassword"><value>password</value></property>
        </bean>

    <bean id="userSearch"
class="org.acegi.security ldap.search.FilterBasedLdapUserSearch">
        <constructor-arg value="ou=People, dc=acme, dc=org"/>
        <constructor-arg value="(uid={0})"/>
        <constructor-arg ref="initialDirectoryContextFactory" />
        <property name="searchSubtree" value="true"/>
    </bean>

    <bean id="userManager"
        class="org.mule.galaxy.security ldap.LdapUserManager"
        init-method="initialize">
        <property name="initialDirectoryContextFactory"
ref="initialDirectoryContextFactory"/>
        <property name="persisterManager" ref="persisterManager" />
        <property name="userSearch" ref="userSearch" />
        <property name="userManager" ref="userManager" />
        <property name="ldapUserMetadataDao" ref="ldapUserMetadataDao" />

        <!-- Configure these two properties -->
        <property name="userSearchBase" value="ou=People, dc=acme, dc=org"/>
        <property name="userSearchAttributes">
            <map>
                <entry key="objectclass" value="inetOrgPerson"/>
            </map>
        </property>
    </bean>

    <bean id="ldapAuthoritiesPopulator"
class="org.mule.galaxy.security ldap.LdapAuthoritiesPopulator">
        <constructor-arg ref="initialDirectoryContextFactory" />
        <constructor-arg value="ou=Groups, dc=acme, dc=org" />
        <property name="groupSearchFilter" value="uniqueMember={0}"/>
        <property name="searchSubtree" value="true"/>
        <property name="rolePrefix" value=""/>
        <property name="convertToUpperCase" value="false"/>
        <property name="accessControlManager" ref="accessControlManager"/>
    </bean>

</beans>

```

Following are the properties you set in this file:

- The `managerDn` property of the `initialDirContextFactory` bean. This is the distinguished name (DN) of the user you will use to log in to the LDAP server.
- The `managerPassword` property of the `initialDirContextFactory` bean. This is the password of the user you will use to log in to the LDAP server.
- The first `<constructor-arg>` of the `userSearch` bean. This is the base context in which Tcat Server will search for users.
- The second `<constructor-arg>` of the `userSearch` bean. This is a filter expression used to find entries that match a particular user name. For example, `(uid={0})` would look for an entry where the `uid` attribute matches the user name.
- The `userSearchBase` property of the `userManagerTarget` bean. This is the base context in which Tcat Server will search for users.
- The `userSearchAttributes` property of the `userManagerTarget` bean. These attributes are used to search for users in the LDAP server.
- The second `<constructor-arg>` of the `LdapAuthoritiesPopulator` bean. This is the DN of the context you will use to search for roles the user belongs to.
- The `groupSearchFilter` property of the `LdapAuthoritiesPopulator` bean. This is an expression that finds roles. For instance, `(uniqueMember={0})` searches for roles inside of the `groupSearchBase` that have an attribute `uniqueMember` where one of the values is the user name.

Working with Server Profiles

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 also add files from the repository, such as configuration files like `server.xml`. When you add files, you add all files in the workspace you specify. For information on adding workspaces and files to the repository, see Chapter 6, “[Working with the Repository](#).”

► **To create or edit a profile:**

- 1 On the Administration tab, click **Server Profiles**, and then click **New**.
- 2 Enter a unique name for the profile and optionally a description.
- 3 Click in the Workspace box and select the workspace that contains the files to add to this profile (see Chapter 6, “[Working with the Repository](#),” for instructions on adding a workspace and files).
- 4 For each file in the workspace, specify which version you want to use, or leave the default “Use Latest” selected.
- 5 To remove the files from this profile, select the “none” workspace.
- 6 To add an environment variable to the profile, click **New Variable**, and then double-click the placeholder name and value and enter the correct information.
- 7 To remove an environment variable from the profile, select it in the table and click **Remove Variable**.
- 8 When you are ready to save the profile and publish its settings and files to all servers that are already set to use this profile, click **Save**. To revert your changes, click **Cancel**.

If a server profile contains any environment variables that affect the startup of Tomcat, you should now restart the Tomcat instances that use that profile by selecting them on the Servers tab and clicking **Restart**.

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 Tcat Server to restart Tomcat.

To apply the profile to one or more servers, select the servers on the Servers tab and click **Set Profile**. You can also set the profile when editing a server’s details. For more information, see “Working with Servers.”

To remove a profile, select it on the Server Profiles screen and click **Delete**. The environment variables and files on the servers that used that profile will remain intact even after deleting the profile.

Creating Remote Workspaces

Remote workspaces are workspaces that exist in another Tcat Server instance. Tcat Server allows you to attach these workspaces to your local Tcat Server instance so you can browse them and perform actions such as replication.

Note Searching across remote workspaces is not currently supported.

► **To attach a remote workspace:**

- 1 On the Administration tab, click **Remote Workspaces** in the left navigation pane, and then click **New**.
- 2 Enter the information described below.
- 3 Click **Save**.

Field	Description
Local Workspace Name	The name that will appear locally for the remote workspace.
Attach to Workspace	The local workspace in which you are attaching the remote workspace. As you type a name, local workspaces will be listed. To create a top-level workspace, leave this field blank.
URL	The URL of the remote workspace's Atom feed, using the format <code>http://HOST/api/registry/PATH</code> , where <code>HOST</code> is the remote Tcat Server instance and <code>PATH</code> is the workspace, such as <code>http://acme.com:8080/api/registry/Services/Accounting</code>
Username	The user name in the remote Tcat Server instance that will be used to access the information.
Password	The password for the above user.

Monitoring Repository Activity

An important aspect of governance is the logging of all relevant changes inside the repository. Activities such as the addition of new artifacts are all logged inside Tcat Server. You can easily search and view these updates by clicking the Activity link on the Administration tab.

You can filter the activity log by date, by the user who made the changes, and by artifacts containing specific text or of certain types. You can also type the name of a specific artifact in the Relating to box to see activity for just that item. Lastly, you can specify the maximum number of results you want to return. After specifying the filter conditions you want, click **Search**. To clear the filter and see all activity, click **Reset**.

Applying Your Tcat Server License

When you purchase a Tcat Server license, you apply it to your instance of Tcat Server as follows:

- 1 Copy the file to the same server where Tcat Server is running.
- 2 In the Tcat Server Administration Console, click **Administration**, and then click **License**.
- 3 Click **Browse** and navigate to the location of the file.
- 4 Click **Submit**.

Scheduling the Usage Statistics Script

By default, instances of Tcat Server run the Usage Statistics script every thirty minutes. This script sends a randomly generated ID to enable MuleSoft to track the number of Tcat Server administration console instances in use. It does **not** contain any identifying information such as IP addresses, and it does **not** track the number of Tcat Server agents that are installed (just administration console instances). If you are behind a firewall, this script does not do anything, and it does not generate any errors or disrupt service in any way.

If you want to change or disable this script, do the following:

- 1 On the Administration tab in the Tcat Server administration console, click **Utility** in the left-hand navigation pane, and then click **Scheduler**.
- 2 Click **Usage Statistics Job**.
- 3 To change the frequency of the pings (default is every 30 minutes), change the cron command (see “[Cron Command Syntax](#)” on page 104), and then click **Save**.
- 4 If you do not want this script to run, click **Delete**.

Working with Logs

You can monitor the logs for a Tcat Server instance by viewing the server's details on the Servers tab and then clicking the Logs tab.

Following are the logs generated by Tomcat for Tcat Server:

This file...	Logs events related to...
tcat-agent-app. log	Running the Tcat Server agent, such as starting the Quartz scheduler and loading bean definitions.
tcat-agent-exec-output. YYYY-MM-DD. log	Output from commands and processes spawned by the agent. Includes STDOUT, STDERR, and info on when a command was executed and by which shell account.
tcat-agent-sys. YYYY-MM-DD. log	Running the Tcat Server agent in the container, such as initializing servlets.
tcat-console-app. log	Running the Tcat Server console application, such as indexing, loading plugins, and starting the repository.
tcat-console-sys. YYYY-MM-DD. log	Running the Tcat Server console in the container, such as initializing servlets.
tcat-file-audit. log	File API operations on the server (see below for details).
tcat-shell-scripts. log	Script activity on the server (see below for details).
mule-ibears. log	Mule iBeans activity.

For information on logging in Tomcat, see the Apache Logging FAQs at <http://wiki.apache.org/tomcat/FAQ/Logging#Q1>.

File Audit Log

One of the logs listed is the file audit log (tcat-file-audit. log), which logs every file API operation on the server. Entries in this log are prefixed by TOMCAT_FILE_AUDIT. For example, the following entries indicate that the `list()` and `download()` methods were called on files on this server:

```
[10-21 17:20:56] INFO TOMCAT_FILE_AUDIT [http-51443-1]: list() |  
C:\projects\galaxy-ee\trunk\tcat\provision\target\catalina-9003\  
[10-21 17:21:05] INFO TOMCAT_FILE_AUDIT [http-51443-1]: download() |  
C:\projects\galaxy-ee\trunk\tcat\provision\target\catalina-9003\.webapps\agent\WEB-INF\agent.properties
```

Shell Script Log

The Logs tab also displays the shell script log (tcat-shell-scripts.log) for any scripts you've created in the Tcat Server administration console (see “[Creating a Script](#)” on page 102). The default logging level for the shell script log is INFO, which you can modify in the webapps/console/WEB-INF/classes/log4j.properties file. You can modify the admin.shell-scripts category to change the logging output for all admin console shell scripts, or to modify the output for a single script, modify the admin.shell-scripts.[*scriptName*] category, where *scriptName* is the name of the script with any spaces escaped by a backslash, such as admin.shell-scripts.[My\ Script].

Note If you enter a script in the admin shell and click Evaluate before saving the script, the logging category for that script is admin.shell-scripts.[\$shell].

Appendix A

Using the Maven Publishing Plug-in

This appendix describes how to use the Maven Publishing plug-in to automatically upload your application to the repository. It includes the following sections:

- **“Overview”** on page 125
- **“Adding the Tcat Server Repository to the POM”** on page 125
- **“Configuring the Plug-in”** on page 126
- **“Configuring Security Information”** on page 127
- **“Additional Configuration Options”** on page 127

Overview

If you create a Maven project for your application, you can use the Galaxy publishing plug-in for Maven to easily publish your project and/or its resources to a Tcat Server repository. To use the plug-in, you add the necessary Maven repositories to your Maven POM, and then configure the plug-in using the `<configuration>` tags in your POM. The rest of this appendix describes these tasks.

After using the plug-in to update your applications, you can schedule a script to listen for these updates and automatically redeploy their packages. For an example, see **“Automatically Redeploying After an Artifact Update”** on page 106.

Adding the Tcat Server Repository to the POM

The first step is to add your Tcat Server repository (or snapshot repository) to your Maven POM. The Tcat Server repository includes the MuleForge repository, the Mule Dependencies repository, and the Apache Incubating repository. You add these to the `<plugins>` section so that Maven knows where to get the required files:

```

<pluginRepositories>
  <pluginRepository>
    <id>muleforge</id>
    <url>http://repository.muleforge.org</url>
    <name>MuleForge Repository</name>
  </pluginRepository>

  <pluginRepository>
    <id>mule.dependencies</id>
    <url>http://dist.codehaus.org/mule/dependencies/maven2</url>
    <name>Mule Dependencies Repository</name>
  </pluginRepository>

  <!-- For the Abdera dependencies -->
  <pluginRepository>
    <id>apache-incubating</id>
    <name>Apache Incubating Repository</name>
    <url>http://people.apache.org/repos/m2-incubating-repository</url>
  </pluginRepository>
</pluginRepositories>

```

Configuring the Plug-in

After adding the Maven repositories to the POM, you configure the generated WAR from Maven to your Tcat Server repository:

```

<build>
  <plugins>
    <plugin>
      <groupId>org.mule.galaxy</groupId>
      <artifactId>galaxy-maven-publish-plugin</artifactId>
      <version>2.0-M3</version>
      <configuration>
        <url>http://localhost:8080/console/api/registry/Applications</url>
        <username>admin</username>
        <password>admin</password>

        <!-- Publish generated WAR -->
        <includes>
          <include>target/*-${version}.war</include>
        </includes>
      </configuration>
    </plugin>
  </plugins>
</build>

```

```

    <execution>
      <id>publish-artifacts</id>
      <phase>package</phase>
      <goals>
        <goal>execute</goal>
      </goals>
    </execution>
  </executions>
</plugin>
...

```

Configuring Security Information

If you do not want to include the user name and password in your POM, you can put it in your `~/.m2/settings.xml` file. The file will look like this:

```

<settings>
  <servers>
    <server>
      <id>myServer</id>
      <username>admin</username>
      <password>admin</password>
    </server>
  </servers>
</settings>

```

To use this security information, add the following tag to your plug-in configuration, replacing `myServer` with the same ID you used in the `settings.xml` file above:

```

<configuration>
  ...
  <serverId>myServer</serverId>
  ...
</configuration>

```

Additional Configuration Options

For more options you can use in the `<configuration>` section of the POM, see the generated documentation page.

Index

A

- Admin shell 102
- Apache Tomcat
 - about 11
- APIs, programmatic access to Tcat Server 110
- applications
 - adding to the repository 99
 - security for 100
 - uploading 52
 - uploading via Maven 125
 - viewing 57
- artifacts
 - about 98
 - modifying 98
 - removing 101
 - security for 100
- attributes 57, 58
- Attributes view 64
- audience for this guide 8
- authentication
 - enabling through LDAP 115
- automating tasks 102

C

- connections 64
- context attributes 64
- Context Descriptor view 65
- context path 65
- context.xml 65
- copying workspaces 110
- cron command syntax 104

D

- deleting server groups 50
- Deployment Descriptor view 65
- documentation
 - typographic conventions 8
- downloading Tomcat 26

E

- environment variables
 - managing in server profiles 119
 - setting manually for a server 55
- example scripts 106
- extending Tcat Server 102

F

- file audit log 122
- Filters view 65

G

- Groovy scripts 102
- groups
 - see* server groups

I

- Info tab 99
- installing multiple Tcat Server instances 21
- installing Tcat Server
 - typical path 12, 17
- Items tab 98

J

- JAVA_OPTS setting 28
- JMX agent
 - enabling 59
- JNDI resources 64
- JSPs view 64

K

- knowledge base 9

L

- LDAP
 - enabling authentication through 115
- ldap.xml 116
- license
 - applying 121
- logs 122
 - viewing 58

M

- mappings
 - filter 65
 - servlet 65
- Maven Publishing plug-in 125
- metadata
 - adding 99
- modifying packages 53
- monitoring repository activity 120
- monitoring servers 55
- MuleSoft Technical Support 9
- multiple Tcat Server instances 21
- multi-threaded servlet status 65

N

- notifications
 - subscribing to 100

P

- packages
 - modifying 53
- Parameters view 65
- pool size 64

Q

- quick start 12

R

- registering servers 45
- remote workspaces 120
- replicating workspaces 110
- repository
 - adding applications to 99
 - removing items from 101
 - uploading applications to 52
 - uploading applications via Maven 125
 - viewing contents of 98
- repository activity
 - monitoring 120
- Resources view 64
- REST API 110
- restarting a server 27, 55
- roles
 - about 100
- RSS feeds 100
- running Tcat Server 26

S

- scripting examples 106
- scripts
 - creating 102
 - modifying 103
 - scheduling 103
 - Usage Statistics 121
- security
 - setting 100

- server groups
 - about 46
 - adding servers to 49
 - creating 46
 - deleting 50
 - removing servers from 49
 - renaming 47
- server profiles
 - creating 118
 - deleting 119
 - editing 118
- servers
 - adding to groups 49
 - finding 47
 - monitoring 55
 - registering 45
 - removing from groups 49
 - unregistering 50
- servlet class 65
- servlet context attributes 64
- ServletContext attributes 58
- Servlets view 65
- session attributes 57
- session size 64
- Sessions view 64
- shell script log 123
- starting Tcat Server 26
- stopping Tcat Server 26
- support 9

T

- tasks
 - automating 102
- Tcat Server
 - extending 102
 - overview 10
 - starting 26
 - stopping 26
 - uninstalling 38
- technical support 9
- threads
 - viewing 59

- Tomcat
 - about 11
 - downloading 26
- typographic conventions 8

U

- uninstalling Tcat Server 38
- unregistering servers 50
- uploading applications into the repository 52
- Usage Statistics script 121
- users
 - creating 113
 - deleting 113
 - editing 113

W

- web.xml 65
- WHOIS query 64
- workspaces
 - about 98
 - adding 99
 - modifying 98
 - remote 120
 - removing 101
 - replicating 110



30 Maiden Lane
Suite 500
San Francisco, CA 94108
Phone: 877-MULE-OSS (877-685-3677)
Fax: 415-227-0842
www.mulesoft.com