



Purpose

This document will cover how to monitor Apache Tomcat servers using the Apache Tomcat wizard and `check_tomcat.jar` plugin within Nagios XI, so that users may be notified when Java applications are behaving unexpectedly.

Target Audience

This document is intended for use by Nagios XI Administrators who want to monitor their Apache Tomcat instances.

Prerequisites

This document assumes you have the following:

- A remote Apache Tomcat server with JMX enabled
- A Nagios XI server with a network route to the Apache Tomcat server

The `check_tomcat.jar` monitoring plugin is executed either on the [Nagios XI server](#) or the [Apache Tomcat server](#). Either method requires some prerequisite steps to be followed first which are outlined below.

Plugin Executed From Nagios XI Server

If you intend to run the plugin from the XI server, you'll need to install Java on the Nagios XI server. At the time of this writing, any Java 7+ implementation should work with the `check_tomcat.jar` plugin, but only Oracle Java and OpenJDK have been tested. The following commands require you to establish a terminal session to your Nagios XI server as the root user.

CentOS / RHEL / Oracle Linux

To install OpenJDK 8 on CentOS / RHEL / Oracle Linux execute the following command:

```
yum install -y java-1.8.0-openjdk-devel
```

Debian 9 / Ubuntu 16,18

To install OpenJDK 8 on Debian 9 / Ubuntu 16,18 execute the following commands:

```
apt-get update
apt-get install -y openjdk-8-jdk
```

Once these steps have been performed please proceed to the [Configuration Wizard](#) section of this document.

Plugin Executed From Remote Tomcat Server

If the plugin is to be remotely executed on the Tomcat server then NCPA will need to be installed on the Tomcat server as per the [Installing NCPA](#) documentation.

Once installed you will need to download the `check_tomcat.jar` plugin to the NCPA's `plugins` folder. The plugin can be downloaded directly from the Nagios XI server, in the following commands replace `xi_address` with the IP address of your Nagios XI server. In a terminal session on the Tomcat server execute the following commands:

```
cd /usr/local/ncpa/plugins/
wget http://xi_address/nagiosxi/includes/configwizards/java-as/plugins/check_jvm.jar
```

The `check_tomcat.jar` is a Java file that NCPA cannot run by default. To have NCPA associate `.jar` files with Java you will need to add a line to the `/usr/local/ncpa/etc/ncpa.cfg` file.

To edit the `ncpa.cfg` file execute the following command:

```
sudo vi /usr/local/ncpa/etc/ncpa.cfg
```

When using vi, to make changes press `i` on the keyboard first to enter insert mode and press `Esc` to exit

insert mode.

Locate the `[plugin directives]` section by typing this command in:

```
/[plugin
```

Scroll down a few lines and find the following line:

```
.py = python $plugin_name $plugin_args
```

Insert the following line after the `.py` line:

```
.jar = java -jar $plugin_name $plugin_args
```

When you have finished, save the changes in vi by typing:

```
:wq
```

and press Enter.

If you have several versions of java installed, you may want to use an absolute path to the binary in place of `java`. If you need to load additional classes for your plugins (not required for JMX over RMI), you may want to specify a classpath using `-cp` between `-jar` and `$plugin_name`.

The last step required is to restart the `ncpa_listener` service. The command to do this may vary depending on your operating system (full details can be found in the [Installing NCPA](#) documentation). In this example to restart the service on CentOS 7 would be:

```
systemctl restart ncpa_listener.service
```

Once these steps have been performed please proceed to the [Configuration Wizard](#) section of this document.

The Apache Tomcat Configuration Wizard

The Apache Tomcat config wizard uses JMX (generally over RMI) to retrieve JVM and system statistics and compare them to the thresholds you set in the wizard. Checks can either be combined into one service or separated.

To begin using the Apache Tomcat configuration wizard, navigate via the top bar to **Configure > Configuration Wizards**. Then, select the **Apache Tomcat** wizard. In the following screenshot you can see how the search field allows you to quickly find a wizard.

The screenshot shows the Nagios XI interface. The top navigation bar includes 'Home', 'Views', 'Dashboards', 'Reports', 'Configure' (circled in blue), 'Tools', 'Help', and 'Admin'. The left sidebar has a 'Configure' dropdown menu with 'Configuration Wizards' circled in blue. The main content area is titled 'Configuration Wizards - Select a Wizard'. Below the title is a search bar with 'Tomcat' entered. A list of icons represents different services, with 'Apache Tomcat' highlighted by a blue box. The description for the Apache Tomcat wizard is: 'Monitor Apache Tomcat JVMs through JMX (uses NCPA or requires Java)'.

Step 1 requires you to provide the details for the Nagios XI server to connect to Tomcat via JMX.

In **Tomcat Server Information**, specify the following:

- **IP Address** is the network address of the Tomcat server
- **Access Tomcat Server via** asks you how to access Tomcat statistics. **JMX** is when Nagios XI connects instead of using NCPA to execute the plugin remotely.
- **NCPA Listener Port** and **NCPA Token** only appear when you select the NCPA access method. You defined these options when installing NCPA earlier.
- **Service URL** is the URL required to form the JMX connection. For JMX over RMI, this will be of the form `service:jmx:rmi:///jndi/rmi://<host>:<port>/jmxrmi`, except in cases where the RMI server is separate from the Tomcat instance.
- **Tomcat Username** and **Tomcat Password** are the credentials required to access the JVM's internal statistics



Configuration Wizard: Apache Tomcat - Step 1

Plugin and Agent Setup

If you use a direct JMX connection between Nagios XI and your application server, you will only need to ensure that your server is remote-JMX-capable. Otherwise, you will need to install NCPA and the Application Server Plugin on your application server.

- [Download and install the latest version of NCPA](#)
- [Install the Java Application Server Plugin](#)
- [Additional documentation for NCPA](#) and for [monitoring application servers](#).

Tomcat Server Information

IP Address:	<input type="text"/>
Access Tomcat Server via:	Remote Agent (NCPA) <input type="button" value="v"/>
NCPA Listener Port	<input type="text" value="5693"/>
NCPA Token	<input type="text"/>
Service URL:	<input type="text"/>
	The full JMX service URL. Ex: "service:jmx:rmi:///jndi/rmi://<host>:<port>/jmxrmi"
Tomcat Username	<input type="text"/>
	The Tomcat user with JMX privileges.
Tomcat Password	<input type="password"/>
	The password for the Tomcat user above.

After making all your selections click **Next** to proceed to **Step 2**.

Step 2 provides you with multiple monitoring options.

In **Remote Host Details** you have the choice of defining the **Host Name** to your requirements. All the services created by this wizard will be assigned to this newly created host. You also have the option to combine the checks into one service.



Configuration Wizard: Apache Tomcat - Step 2

Remote Host Details

IP Address:	<input type="text" value="10.25.9.1"/>
Host Name	<input type="text" value="Tomcat"/>
	The host name you want associated with this check.
Service Description:	<input type="text" value="Tomcat JVM Statistics"/>

- Combine into one service**
This allows you to perform all checks on a single JVM, rather than spooling one JVM per check.

The **Heap-Allocated Memory** and **Non-Heap-Allocated Memory** options are self explanatory, simply check and un-check the relevant boxes to determine which checks to run, and enter your desired warning and critical thresholds.

Heap-Allocated Memory

Measure these statistics in: GiB

- Heap-Allocated Memory**
Measures the memory usage of the entire heap.
⚠️ ⚠️
- Eden Space**
Measures the memory usage of objects which haven't yet seen garbage collection.
⚠️ ⚠️
- Survivor/Tenured Space**
Measures the memory usage of the objects which have survived at least one garbage collection cycle.
⚠️ ⚠️
- Old Gen**
Measures the memory usage of the objects which have been moved out of Survivor Space but are still in use.
⚠️ ⚠️

Non-Heap-Allocated Memory

Measure these statistics in: GiB

- Simple Non-Heap-Allocated Memory**
Measures the memory usage of everything not on the heap.
⚠️ ⚠️
- Code Cache**
Measures the memory usage of the JIT-compiled code.
⚠️ ⚠️
- Compressed Class Space**
Measures the memory usage of the compressed classes in your Tomcat instance.
⚠️ ⚠️
- Metaspace**
Measures the memory usage of the class metadata in your Tomcat instance.
⚠️ ⚠️

For the **Global Request Processors** section, you will also need to provide the name of each request processor you wish to

monitor. These names vary from server to server and from version to version of Tomcat. The correct names can always be found by opening a `jconsole` connection to the Tomcat server, clicking the **MBeans** tab, and then choosing **Catalina > GlobalRequestProcessor**. Each of the resulting subdirectories is named for one of the request processors.

Global Request Processors

To find your global request processors' MBean Names, access the Tomcat instance with `jconsole`.

Request Processor MBean Name	Check Type	Warning Threshold	Critical Threshold
<input type="text"/>	Requests Per Minute <input type="text" value="v"/>	<input type="text"/>	<input type="text"/>

[Add Row](#) | [Delete Row](#)

The **Other System Statistics** options are self explanatory, simply check and un-check the relevant boxes to determine which checks to run, and enter your desired warning and critical thresholds.

Click Next and then complete the wizard by choosing the required options in Step 3 – Step 5.

To finish up, click on **Finish** in the final step of the wizard.

Other System Statistics

- JVM CPU Usage**
Measures the CPU Usage incurred by the JVM alone (values are between 0 and 100)
⚠ 50 % ⓘ 70 %
- System CPU Usage**
Measures the CPU Usage of the system as a whole (values are between 0 and 100).
⚠ 70 % ⓘ 90 %
- Uptime**
Measures the uptime of the JVM in seconds.
⚠ 1500: ⓘ 300:
- Class Count**
Measures the number of currently-loaded classes in the JVM.
⚠ 10000 ⓘ 50000
- Thread Count**
Measures the number of active threads in the JVM.
⚠ 150 ⓘ 190

< Back

Next >

Once the wizard applies the configuration, click the **View status details for <your host>** link to see the new services that have been created.

Finishing Up

This completes the documentation on how to monitor Apache Tomcat in Nagios XI.

If you have additional questions or other concerns, please visit us at our support forums:

<https://support.nagios.com/forum>

The Nagios Support Knowledgebase is also a great support resource:

<https://support.nagios.com/kb>