



Best Practices – Tomcat Performance Tuning for Pentaho

Contents

- Overview..... 1
- Tuning Tomcat for Performance..... 2
 - Choose the Right Connector..... 2
 - Set the Right Thread Pool Size 2
 - Consider Turning-on Compression 3
 - Host Static Content on a Web Server 3
- Conclusion..... 3
- Best Practice Check List..... 4

Overview

We have collected a set of best practice recommendations for you to leverage when you need to do some performance tuning for Tomcat.

Some of the things discussed here for tuning Tomcat include choosing the right connector, setting the correct thread-pool size, and hosting static content on a web server.

The intention of this document is to speak about topics generally; however, these are the specific versions covered here:

Software	Version
Pentaho	6.x, 7.x
Apache Tomcat	7.0, 8.0

More information about the different versions listed above can be found in the [Components Reference](#) documentation on Pentaho's webpage.

Tuning Tomcat for Performance

As with [security](#), the default Tomcat is not configured for performance out-of-the-box, because each environment is different. However, Tomcat does provide a few configuration options to improve performance.

Choose the Right Connector

The configured connectors determine how browser and other clients access the applications deployed within Tomcat. However, Tomcat provides different types of connections, depending on the version. On Tomcat 7, the default Hypertext Transfer Protocol (HTTP) connector is a blocking connector where each worker thread is consumed until its associated connection is complete. However, a non-blocking connector is also available which typically leads to better thread management with longer running requests. Performance tests suggest that the non-blocking connector provides better performance with longer running requests.

For Tomcat 7, use the `org.apache.coyote.http11.Http11NioProtocol` with the connector within the `$tomcat/conf/server.xml`.

```
<Connector connectionTimeout="20000" maxThreads="1000"
  port="8080" redirectPort="8443"
  protocol="org.apache.coyote.http11.Http11NioProtocol" />
```

The default HTTP connection for Tomcat 8 is a non-blocking connector.

Set the Right Thread Pool Size

Each `Connector` can be configured with a thread pool to process the requests it receives. The thread pool should be configured for your anticipated concurrent connections. The default maximum thread pool size within Tomcat 7 is 200.

Consider the following scenario:

If the solution is expecting ~300 new requests per second with each request requiring 2.5 seconds to complete, eventually, there will be ~750 requests that need to be handled simultaneously. The server in this situation should have ~750 processing threads at startup with potentially ~1000 processing threads at maximum to handle excessive loads.

Using a non-blocking Connector may reduce the required amount of threads. However, we recommend starting with several threads to meet the anticipated concurrent connection volume and set the maximum to be the anticipated peak with some buffer.



Only reduce the thread count after analyzing the performance and thread usage of the solution under the anticipated usage load.

For the Pentaho BA Server, clustering the Tomcat instance will allow the thread pool to be virtually distributed across the nodes of the cluster. Thus, each server in the cluster will need less within its thread pool. This may be desired if the workload of a single server is too high.

The example below sets a `Connector` with a minimum thread count of 300 and a maximum of 500. The `prestartminSpareThreads` attribute will cause the connector to prepopulate the thread pool with the configured minimum. The `maxIdleTime` sets the number of milliseconds after which idle threads are released, allowing the thread pool to shrink as volume decreases.

```
<Connector connectionTimeout="20000" port="8080"
  redirectPort="8443"
  maxThreads="500" minSpareThreads="300"
  prestartminSpareThreads="true" maxIdleTime="60000"
  protocol="org.apache.coyote.http11.Http11NioProtocol"/>
```

Consider Turning On Compression

Tomcat is able to compress the content it sends to the browser and other applications. This generally improves performance by reducing the network traffic, but comes at the cost of compressing and decompressing. It is a general practice to turn on compression.



This could degrade performance within fast networks.

We recommend enabling compression during the development phase of the solution, but you should compare it to an uncompressed configuration during testing. It is also possible to set the `compression` attribute to a file size, indicating that only files / content larger than this value will be compressed.

```
<Connector connectionTimeout="20000" port="8080"
  redirectPort="8443" compression="on"
  maxThreads="500" minSpareThreads="300"
  prestartminSpareThreads="true" maxIdleTime="60000" />
```

Host Static Content on a Web Server

Tomcat is designed to deliver dynamic web content. It is a standing practice to front Tomcat with a web server such as Apache Hypertext Transfer Protocol Daemon (HTTPD), allowing the web server to serve the static content. Pentaho does not support deploying any portion of its web applications on a web server; however, it is a recommended practice to store report images or any static content referenced by dashboards on some web server versus within the Tomcat hosting Pentaho. [Use Apache httpd with SSL for Delivering Static Content](#) in the Pentaho documentation has more information.

Conclusion

The guidelines presented in this document are generic in that they were not written with any particular solution in mind. They are considered guidelines and a starting point. These guidelines must be tested and validated within development or QA before being placed into production.

Task Check List

This checklist is designed for you to use while you are thinking about how to secure and tune the performance of Tomcat.

Name of the Project _____

Date of the Review _____

Name of the Reviewer _____

Considerations/Think About:	Response	COMMENTS, WHY?
<i>Did you choose the right connector?</i>	YES ___ NO ___	
<i>Have you set the right thread pool size?</i>	YES ___ NO ___	
<i>Did you consider turning on compression?</i>	YES ___ NO ___	
<i>Are you hosting static content on a web server?</i>	YES ___ NO ___	