

JMeter Plugin

<p>Your browser does not support iframes.</p>

Description / Features

This plugin collects performance tests results from [Apache JMeter](#). [Apache JMeter](#) may be used to test performance both on static and dynamic resources (files, Servlets, Perl scripts, Java Objects, Data Bases and Queries, FTP Servers and more). It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types. You can use it to test your server/script/object behavior under heavy concurrent load.

If it is the first time you have heard of [Apache JMeter](#) then you should try it before using this plugin.

The following metrics are recorded in SonarQube:

- Duration (in seconds)
- Error density (number of errors / number of requests * 100)
- Average response time for requests
- Average response time for transactions
- Successful requests per minute
- Successful requests per minute and user
- Response time deviation for requests
- Response time deviation for transactions
- Number of requests
- Number of transactions
- Successful transactions per minute
- Successful transactions per minute and user

Installation

1. Install the plugin through the [Update Center](#) or download it into the `SONARQUBE_HOME/extensions/plugins` directory
2. Restart the SonarQube server

Quick Guide



New feature in version 0.2

There is now a `jmeter-report-server` embedded inside the plugin, and there is a new configuration option (Local jtl file) where you can set a jtl file path, local to the SonarQube server. If this option is set, no remote jmeter report server is used, and jmeter test results are taken from the local file.

If you want to use JMeter Report Server then:

- Download [jmeter-report-server-0.2-jar-with-dependencies.jar](#)
- Run the server where your JMeter instance is running:

```
> java -Dtestconfig.name.jtlpath=/path/to/file.jtl -jar jmeter-report-server.jar
```

- Go to SonarQube system settings and set host and port for your JMeter Report Server.
- Go to your project settings and set the "configuration key" (this must be the same you set in your property name).
- Run `mvn sonar:sonar`
- Add JMeter widget to your dashboard and your JMeter metrics will be displayed 😊



JTL files must be provided as a result of JMeter tests. JMeter Report Server is just to make JTL files available remotely.

Usage

JMeter

JMeter installation is very easy. JMeter is developed with Java and it is distributed as a compressed file.

You can download it from <http://jakarta.apache.org/jmeter> and then run the provided script (bin/jmeter).

Recording a test in JMeter

You can find a step by step tutorial in JMeter's website: http://jakarta.apache.org/jmeter/usermanual/jmeter_proxy_step_by_step.pdf

It describes how to record a test plan using the JMeter proxy.

You can also take a look at this video tutorial to see how to record a test with JMeter:

The SonarQube JMeter plugin uses the results datafile generated by JMeter (file with extension .JTL) so remember to configure your test plan to store the results in a single JTL file.

In this video we show you how to set this JTL file:

JMeter Report Server

JMeter Report Server is a server that generates performance reports from JMeter results. The SonarQube JMeter Plugin uses this server to get the performance metrics from the projects.

It must be installed on the same machine with JMeter tool. The server will read the JTL files to provide all the data to SonarQube.

JTL files must be provided as a result of JMeter tests. The JMeter Report Server is needed just to make JTL files available remotely.

JMeter Report Server Configuration

JMeter Report Server needs to know where the JTL files are located. You can set your test configuration when running the server through command line system properties:

```
java -Dtestconfig.name1.jtlpath=/path/to/file1 -jar jmeter-report-server.jar
```

where name is the test configuration that you will use in SonarQube JMeter Plugin project settings.

Or you can also set this location by creating a property file "jmeter-report-server.properties" with all your test configurations:

```
testconfig.name1.jtlpath=/path/to/file1
testconfig.name2.jtlpath=/path/to/file2
...
testconfig.nameM.jtlpath=/path/to/fileN
```

This is useful if you have multiple projects in your JMeter environment and you want to manage all your tests with a single JMeter Report Server.

Port and maxConnections are now configurable properties and they can be set as system properties (using -D in the start command) or as normal properties in the jmeter-report-server.properties file as shown below.

```
# Using default port
port=4444

# Unlimited connections
maxConnections=0
```

Here you have a test configuration example:

```
testconfig.iceweb.jtlpath=/home/carlos/jmeter_work/iceweb.jtl
```

This configuration will set JTL file "/home/carlos/jmeter_work/iceweb.jtl" to the project "iceweb".

"iceweb" will be the name for your test configuration. SonarQube plugin settings will use that name.

JMeter Report Server Installation

Server is package in a single jar with all dependencies, so just download JMeter Report Server and put it wherever you want 😊

Running JMeter Report Server

You can start the server with this command:

```
java -Dtestconfig.name1.jtlpath=/path/to/file1 -jar jmeter-report-server.jar
```

NOTE: replace "jmeter-report-server.jar" with the name of the file downloaded and testconfig property with your project name and the path to the JTL file.

SonarQube JMeter Plugin

Configuration

This video shows how to configure JMeter plugin in SonarQube:

- Project must exist in SonarQube before using JMeter plugin
- JMeter Report Server must be up and running if using remote configuration, and it must have a test configuration with the same name you set in SonarQube JMeter plugin settings.
- JTL file must exist in the host that is running JMeter Report Server.

Since version 0.2, there is a jmeter-report-server embedded inside the SonarQube JMeter plugin, and there is a new configuration option (Local jtl file) where you can set a jtl file path, local to the sonar server. If this option is set, no remote jmeter report server is used, and jmeter test results are taken from the local file.

Now it is possible to configure units of time in your global or project settings. You can set seconds, minutes or hours.

Add the JMeter widget to your dashboard and next time you run `mvn sonar:sonar` you will retrieve performance metrics from JMeter, and SONAR will show them.

Usage with maven-jmeter-plugin

Since version 0.2, the plugin can be used together with [Maven JMeter Plugin](#) and [Cargo](#) to perform the tests and collect performance metrics in just one Maven goal:

```
mvn cargo:start org.apache.jmeter:maven-jmeter-plugin:jmeter sonar:sonar
```

You have to configure your project to be able to do that:

- Include your `jmeter.properties` and all your JMeter tests in "`src/test/jmeter`"
- Configure your new maven plugins in your project and set "`jmeter-reports`" as the your report directory.

```

<!-- Plugin to launch your JMeter tests in Maven -->
<plugin>
  <groupId>org.apache.jmeter</groupId>
  <artifactId>maven-jmeter-plugin</artifactId>
  <version>1.0-GA</version>
  <configuration>
    <reportDir>${project.build.directory}/jmeter-reports</reportDir>
  </configuration>
</plugin>

<!-- Plugin to start a container and be able to run your webapp and perform your
JMeter tests
  This example uses Tomcat 7 as the container. -->

<plugin>
  <groupId>org.codehaus.cargo</groupId>
  <artifactId>cargo-maven2-plugin</artifactId>
  <version>1.1.0</version>
  <configuration>
    <container>
      <containerId>tomcat7x</containerId>
      <zipUrlInstaller>

<url>http://archive.apache.org/dist/tomcat/tomcat-7/v7.0.12/bin/apache-cat-7.0.12.zip<
/ur
      </zipUrlInstaller>
    </container>
  </configuration>
</plugin>

```

To get sonar-jmeter-plugin working in maven side, there must not be any configuration values set in SonarQube (all of them must be blank).

Known Limitations

- You can only use one JTL file per project.

Change Log

Version 0.2 (8 issues)

T	Key	Summary
	SONARPLUGINS-1339	Apache JMeter 2.5 support
	SONARPLUGINS-1274	Allow to set jtl path as a classpath path
	SONARPLUGINS-1271	Sometimes sockets are not closed in the server
	SONARPLUGINS-1270	Limit the number of connected clients
	SONARPLUGINS-1259	Allow jmeter-report-server port to be configurable
	SONARPLUGINS-1198	Implement JMeterPostJob for getting performance metrics in maven side parsing reports generated by maven-jmeter-plug
	SONARPLUGINS-1018	Configuring unit

8 issues

Version 0.1 (5 issues)

T	Key	Summary
	SONARPLUGINS-983	Update documentation
	SONARPLUGINS-982	Plugin usability: command line parameter for JTL file
	SONARPLUGINS-956	Code best practices
	SONARPLUGINS-913	Plugin documentation
	SONARPLUGINS-906	Upload code to SVN

5 issues