


# JMeter Plugin

 Do not use this plugin anymore. Use [Performance plugin](#) instead.

Due to the new ability of this plugin of handling junit reports as well as jmeter ones, JMeter-plugin has been renamed to [Performance plugin](#)

## Introduction

This plugin allows you to capture report from [JMeter](#). Hudson will generate graphic charts with the trend report of performance and robustness.

Release 0.3.0 includes the feature of setting the final build status as good, unstable or failed, based on the reported jmeter error percentage.

## Usage with Freestyle project

1. You can use ant and jmeter task for ant to launch tests on your web application
2. Your ant file may be similar to something like that:

```
<project default="all">
  <!-- Define an environment variable pointing to JMETER folder or change this -->
  <property environment="env"/>
  <property name="jmeter-home" location="${env.JMETER_DIR}"/>

  <!-- ant-jmeter.jar comes with jmeter, be sure this is the release you have -->
  <path id="ant.jmeter.classpath">
    <pathelement
      location="${jmeter-home}/extras/ant-jmeter-1.0.9.jar" />
  </path>

  <taskdef
    name="jmeter"
    classname="org.programmerplanet.ant.taskdefs.jmeter.JMeterTask"
    classpathref="ant.jmeter.classpath" />

  <target name="clean">
    <delete dir="results"/>
    <delete file="jmeter.log"/>
    <mkdir dir="results/jtl"/>
    <mkdir dir="results/html"/>
  </target>

  <target name="test" depends="clean">
    <jmeter
      jmeterhome="${jmeter-home}"
      resultlogdir="results/jtl">
      <testplans dir="test/jmeter" includes="*.jmx"/>
    </jmeter>
  </target>

  <!-- This is not needed for the plugin, but it produces a nice html report
  which can be saved usin hudson's archive artifact feature -->
  <target name="report" depends="test">
    <xslt
      basedir="results/jtl"
      destdir="results/html"
      includes="*.jtl"
      style="${jmeter-home}/extras/jmeter-results-detail-report_21.xsl"/>
  </target>

  <target name="all" depends="test, report"/>
</project>
```

## Hudson Configuration

In Hudson, configure the files or the repository containing the files to be parsed by JMeter plugin.

If you have are using version 0.3.0, you can configure the error percentage thresholds which would make the project unstable or failed.

Publish JMeter test result report

JMeter report

Specify the path to the JMeter report files, relative to [the workspace root](#). The path is a list of semicolon separated values pointing either to a directory or a file, e.g. "target/jmeter-reports";target/jmeter-reports/perf\_continue\_test1.xml".

JMeter threshold ● Unstable ● Failed

Thresholds:  %  %

Specify the error percentage threshold that set the build unstable or failed (a value of 0 means: don't use this threshold).

## Other Articles

- [Performance tests with Hudson, Maven, and JMeter](#)

## Compiling

To use the latest plugin release, you need to download, compile and install by hand. To do it, you need subversion, maven and java installed in your computer.

```
$ svn checkout https://svn.dev.java.net/svn/hudson/trunk/hudson/plugins/jmeter jmeter (username is 'guest' and password 'guest')
$ cd jmeter
$ mvn package
$ cp target/jmeter.hpi <path_to_hudson>/data/plugins
```

## Changelog

### v0.3.0

- Added the possibility of using more than one jmeter-report file. In order to keep the last version behaviour, the trend graphs will appear on the front page of the Hudson job if only one report file is present, when several report files are present the graphs will appear only when you click on "Jmeter trend".
- Added a configuration page for the graphs, accessible from the "Jmeter trend" page. It permits to show only the n last builds or the builds within a range of dates, the configuration is saved in a cookie.
- Added threshold parameters in order to report the build process as unstable or failed based on the error percentage.
- Fixed some bugs related with the parsing of previous-builds not configured for jmeter tasks.
- Improvement of the plugin's performance when generating the graphs.
- Added support for non-http XML report files.

**NOTE:** There is an incompatibility with the previous version of the plugin, due to the way the JMeter result files are handled:

- The Maven JMeter plugin handles the test result in this way now:

\* A test myJmeterTest.jmx produces a result file myJmeterTest-yyMMdd.xml where the date is the execution date of the test.

- The last released version (0.2) of the Hudson JMeter plugin only handles one JMeter test:

\* When the build finishes the JMeter result file myJmeterTest-yyMMdd.xml is copied to the root of the build directory and renamed to "jmeter.xml".

- But version 0.3.0 introduces the possibility to handle several JMeter tests :

\* When the build finishes the JMeter result files myJmeterTest-yyMMdd.xml and myOtherJmeterTest-yyMMdd.xml are copied to the build/jmeter-reports directory and renamed to "myJmeterTest.xml" and "myOtherJmeterTest.xml"

\* The result file name is used to keep track of each test when building the trend graphs, we cannot use anymore the hardcoded file name "jmeter.xml"

\* Unless the unique JMeter test file is called jmeter.jmx, there is no way to make the 2 versions of the plugin compatible.

- To keep the current test results obtained with the 0.2 plugin one has to apply the following migration procedure :

\* Rename the result file jmeter.xml found at the build root to "nameOfTheJmeterTest.xml"

\* Move the result file "nameOfTheJmeterTest.xml" to the build/jmeter-reports directory

\* The procedure has to be performed for all the existing builds

### v0.2

handling jmeter log format v 2.0

### v0.1

first release