APACHE JMETER

asynchronous javascript and xml

tutorialspoint

SIMPLYEASYLEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

About the Tutorial

jMeter is an open source testing software. It is 100% pure Java application for load and performance testing.

jMeter is designed to cover various categories of tests such as load testing, functional testing, performance testing, regression testing, etc., and it requires JDK 5 or higher.

This tutorial provides an in-depth coverage of jMeter framework including its test plans, listeners, functions, and regular expressions.

Audience

This tutorial is designed for software professionals in the domain of Testing who are required to test enterprise-level applications for robustness and reliability.

Prerequisites

Before proceeding with this tutorial, you should have a basic understanding of Java programming language. As you are going to use jMeter to perform all types of testing (regression, functional, load, performance, etc.) at different stages of a Java project, it is recommended that you have a good grip over software development and software testing processes.

Copyright & Disclaimer

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

1. JMETER – OVERVIEW

Before going into the details of JMeter, let us first understand a few jargons associated with the testing of any application.

- **Performance Test**: This test sets the best possible performance expectation under a given configuration of infrastructure. It also highlights early in the testing process if any changes need to be made before the application goes into production.
-
- **Load Test**: This test is basically used for testing the system under the top load it was designed to operate under.
-
- **Stress Test**: This test is an attempt to break the system by overwhelming its resources.

What is JMeter?

JMeter is a software that can perform load test, performance-oriented business (functional) test, regression test, etc., on different protocols or technologies.

Stefano Mazzocchi of the Apache Software Foundation was the original developer of JMeter. He wrote it primarily to test the performance of Apache JServ (now called as Apache Tomcat project). Apache later redesigned JMeter to enhance the GUI and to add functional testing capabilities.

JMeter is a Java desktop application with a graphical interface that uses the Swing graphical API. It can therefore run on any environment / workstation that accepts a Java virtual machine, for example: Windows, Linux, Mac, etc.

The protocols supported by JMeter are:

- Web: HTTP, HTTPS sites 'web 1.0' web 2.0 (ajax, flex and flex-ws-amf)
- Web Services: SOAP / XML-RPC
- Database via JDBC drivers
- Directory: LDAP
- Messaging Oriented service via JMS
- Service: POP3, IMAP, SMTP
- FTP Service

JMeter Features

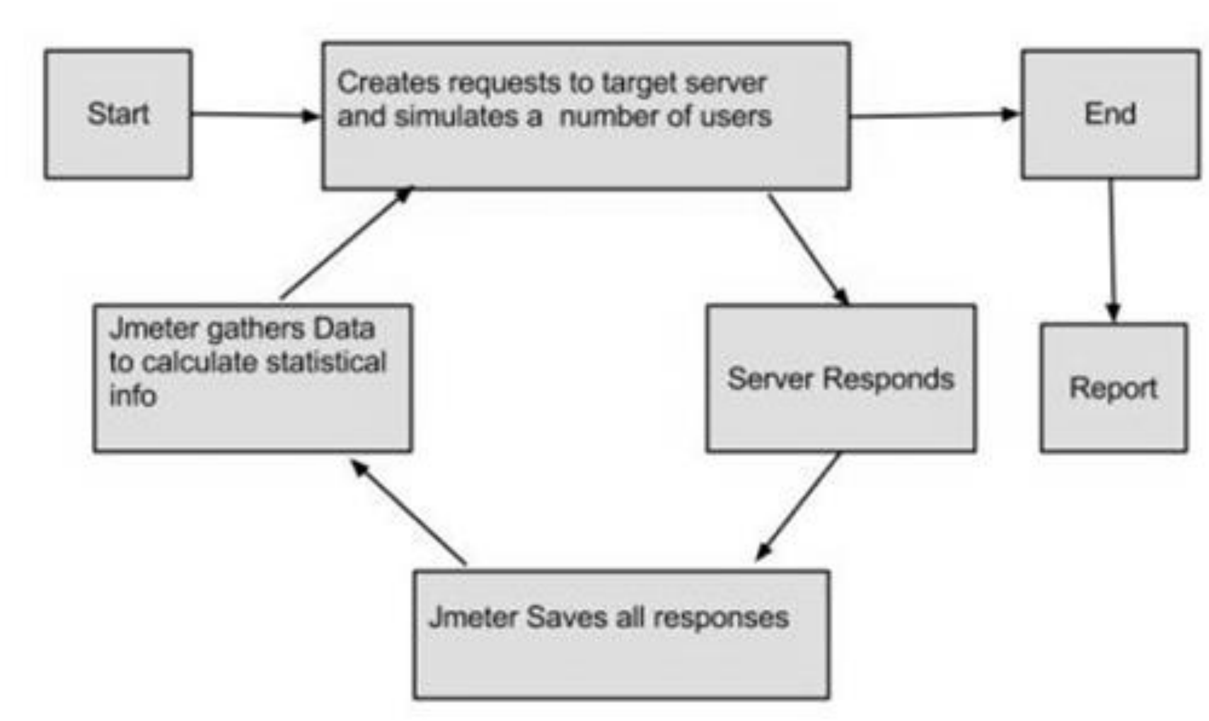
Following are some of the features of JMeter:

- Being an open source software, it is freely available.
- It has a simple and intuitive GUI.
-
- JMeter can conduct load and performance test for many different server types: Web - HTTP, HTTPS, SOAP, Database via JDBC, LDAP, JMS, Mail - POP3, etc.
-
- It is a platform-independent tool. On Linux/Unix, JMeter can be invoked by clicking on JMeter shell script. On Windows, it can be invoked by starting the jmeter.bat file.
-
- It has full Swing and lightweight component support (precompiled JAR uses packages javax.swing.*).
-
- JMeter stores its test plans in XML format. It means you can generate a test plan using a text editor.
-
- Its full multi-threading framework allows concurrent sampling by many threads and simultaneous sampling of different functions by separate thread groups.
-
- It is highly extensible.
-
- It can also be used to perform automated and functional testing of the applications.

How JMeter Works?

JMeter simulates a group of users sending requests to a target server, and returns statistics that show the performance/functionality of the target server/application via tables, graphs, etc.

Take a look at the following figure that depicts how JMeter works:


Start
Creates requests to target server and simulates a number of users
End
Jmeter gathers Data to calculate statistical info
Server Responds
Report
Jmeter Saves all responses

2. JMETER – ENVIRONMENT

JMeter is a framework for Java, so the very first requirement is to have JDK installed in your machine.

System Requirements

JDK	1.6 or above.
Memory	No minimum requirement.
Disk Space	No minimum requirement.
Operating System	No minimum requirement.

Step 1 - Verify Java Installation

First of all, verify whether you have Java installed in your system. Open your console and execute one of the following **java** commands based on the operating system you are working on.

OS	Task	Command
Windows	Open Command Console	c:\> java -version
Linux	Open Command Terminal	$ java -version
Mac	Open Terminal	machine:~ joseph$ java -version

If you have Java installed in your system, you would get an appropriate output based on the OS you are working on.

OS	Output
Windows	java version "1.7.0_25"

	Java(TM) SE Runtime Environment (build 1.7.0_25-b15) Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)
Linux	java version "1.7.0_25" Java(TM) SE Runtime Environment (build 1.7.0_25-b15) Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)
Mac	java version "1.7.0_25" Java(TM) SE Runtime Environment (build 1.7.0_25-b15) Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)

If you do not have Java installed, install the Java Software Development Kit (SDK) from http://www.oracle.com/technetwork/java/javase/downloads/index.html. We are assuming Java 1.7.0_25 as the installed version for this tutorial.

Step 2: Set Java Environment

Set the **JAVA_HOME** environment variable to point to the base directory location, where Java is installed on your machine. For example:

OS	Output
Windows	Set the environment variable JAVA_HOME to C:\Program Files\Java\jdk1.7.0_25
Linux	export JAVA_HOME=/usr/local/java-current
Mac	export JAVA_HOME=/Library/Java/Home

Append Java compiler location to System Path.

OS	Output
Windows	Append the string; C:\Program Files\Java\jdk1.7.0_25\bin to the end of the system variable, Path.

Linux	export PATH=$PATH:$JAVA_HOME/bin/
Mac	not required

Verify Java Installation using **java -version** command as explained above.

Step 3: Download JMeter

Download the latest version of JMeter from http://jmeter.apache.org/download_jmeter.cgi. For this tutorial, we downloaded *apache-jmeter-2.9* and copied it into C:\>JMeter folder.

The directory structure should look like as shown below:

- apache-jmeter-2.9
- apache-jmeter-2.9\bin
- apache-jmeter-2.9\docs
- apache-jmeter-2.9\extras
- apache-jmeter-2.9\lib\
- apache-jmeter-2.9\lib\ext
- apache-jmeter-2.9\lib\junit
- apache-jmeter-2.9\printable_docs

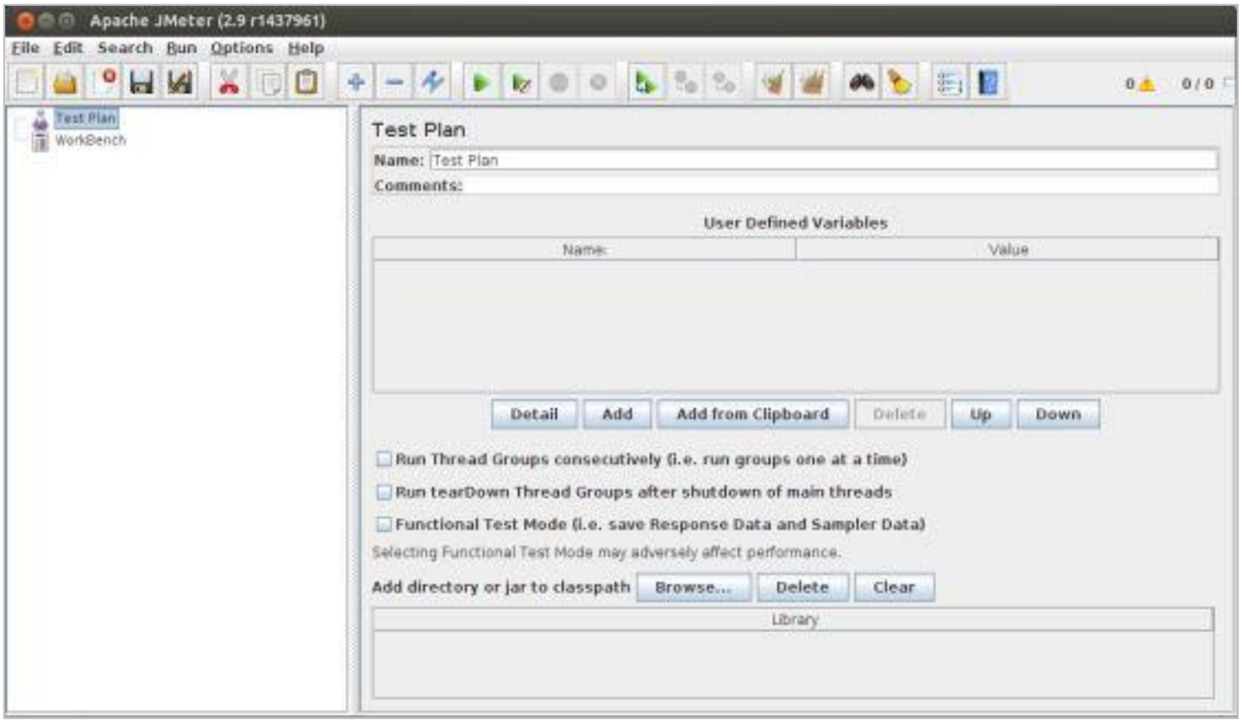
You can rename the parent directory (i.e. apache-jmeter-2.9) if you want, but do not change any of the sub-directory names.

Step 4: Run JMeter

After downloading JMeter, go to the *bin* directory. In this case, it is **/home/manisha/apache-jmeter-2.9/bin**. Now click on the following:

OS	Output
Windows	jmeter.bat
Linux	jmeter.sh
Mac	jmeter.sh

After a short pause, the JMeter GUI should appear, which is a Swing application, as seen in the following screenshot:



Test Plan

Name: Test Plan

Comments:

□ Run Thread Groups consecutively (i.e. run groups one at a time)

□ Run tearDown Thread Groups after shutdown of main threads

□ Functional Test Mode (i.e. save Response Data and Sampler Data)

Selecting Functional Test Mode may adversely affect performance.

Add directory or jar to classpath

This is the main page and the default page of the tool.

3. JMETER – TEST PLAN

What is a Test Plan?

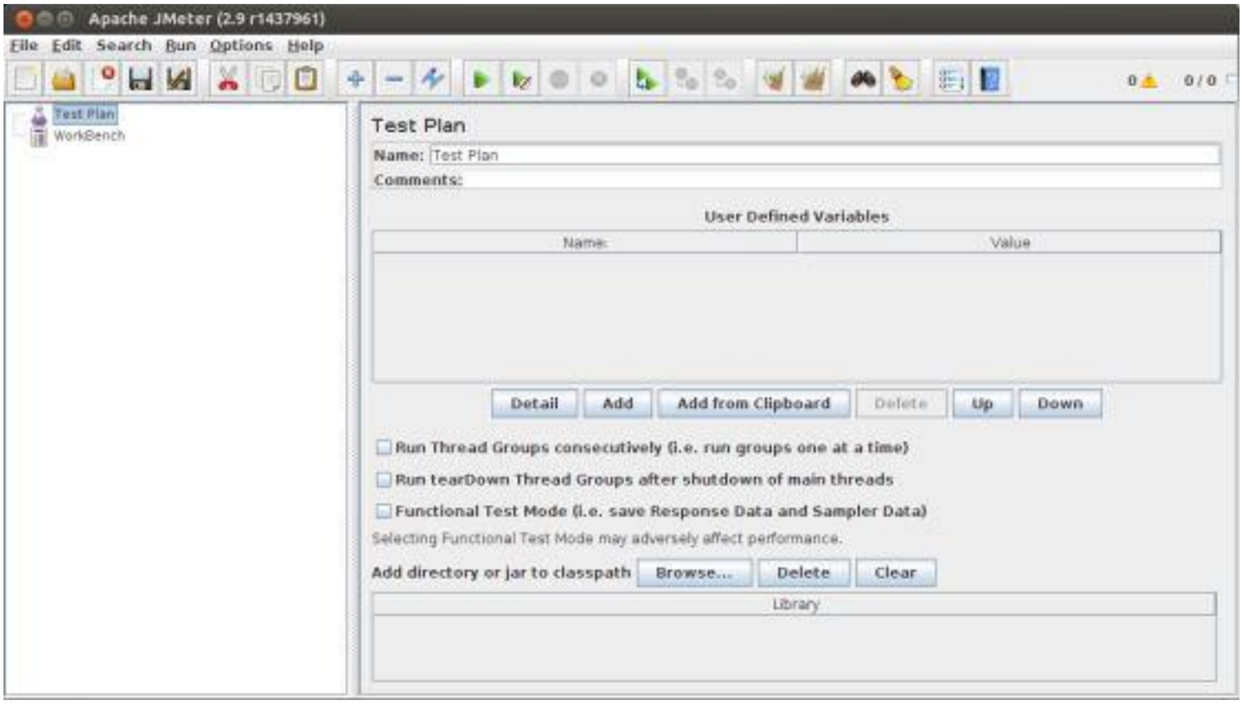
A Test Plan can be viewed as a container for running tests. It defines what to test and how to go about it. A complete test plan consists of one or more elements such as thread groups, logic controllers, sample-generating controllers, listeners, timers, assertions, and configuration elements. A test plan must have at least one thread group.

Writing a Test Plan

Follow the steps given below to write a test plan:

Step 1: Start the JMeter Window

Open the JMeter window by clicking **/home/manisha/apache-jmeter-2.9/bin/jmeter.sh**. The JMeter window will appear as shown below:



Test Plan

Name: Test Plan

Comments:

□ Run Thread Groups consecutively (i.e. run groups one at a time)

□ Run tearDown Thread Groups after shutdown of main threads

□ Functional Test Mode (i.e. save Response Data and Sampler Data)

Selecting Functional Test Mode may adversely affect performance.

Add directory or jar to classpath

This is a plain and blank JMeter window without any additional elements added to it. It contains two nodes:

- **Test Plan node**: It is where the real test plan is kept.
-

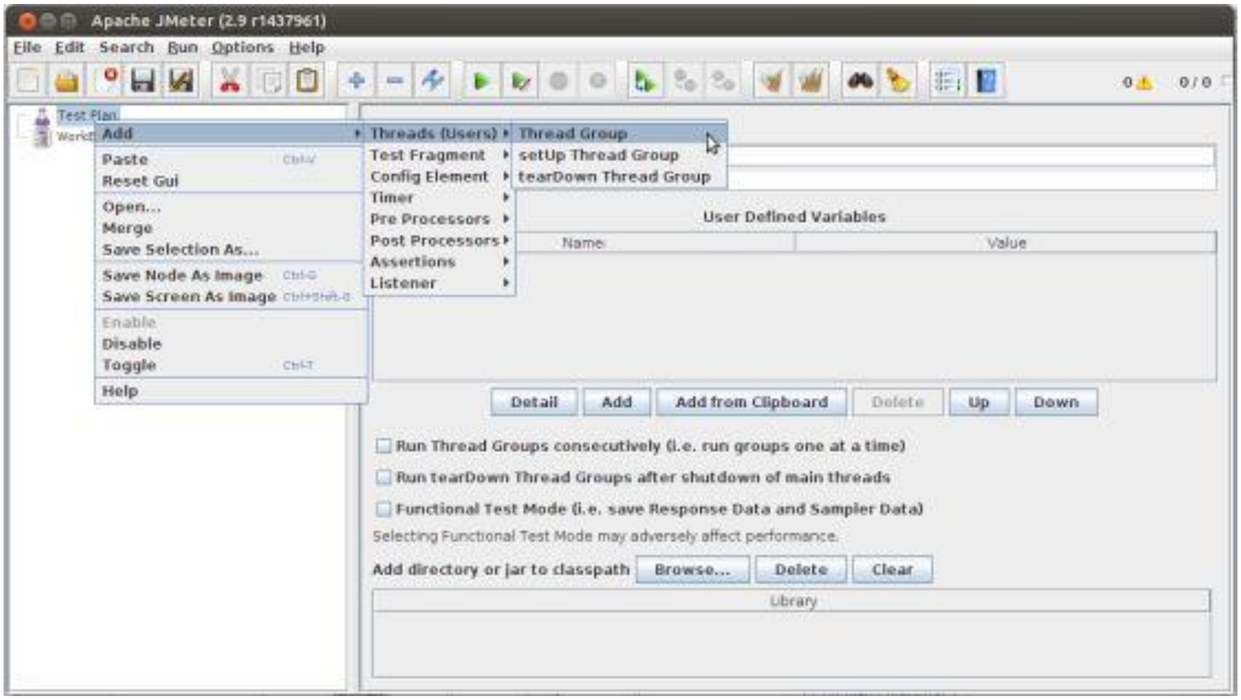
- **Workbench node**: It simply provides a place to temporarily store test elements while not in use, for copy/paste purposes. When you save your test plan, Workbench items are not saved with it.

Step 2: Add/Remove Elements

- Elements (which will be discussed in the next chapter Test Plan Elements) can be added to a test plan by right-clicking on the Test Plan node and choosing a new element from the "add" list.

Alternatively, you can load an element from a file and add it by choosing the "merge" or "open" option.

For example, let us add a Thread Group element to a Test Plan as shown below:



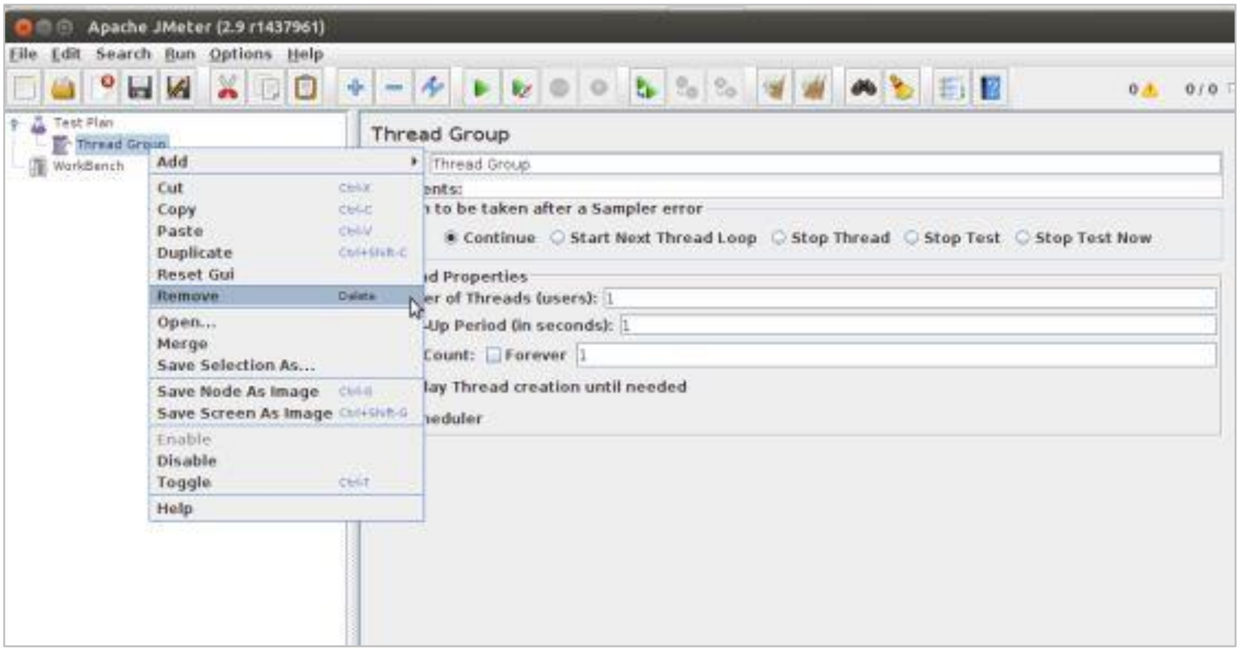
□ Run Thread Groups consecutively (i.e. run groups one at a time)

□ Run tearDown Thread Groups after shutdown of main threads

□ Functional Test Mode (i.e. save Response Data and Sampler Data)

Selecting Functional Test Mode may adversely affect performance.

To remove an element, make sure the element is selected, right-click on the element, and choose the "remove" option.

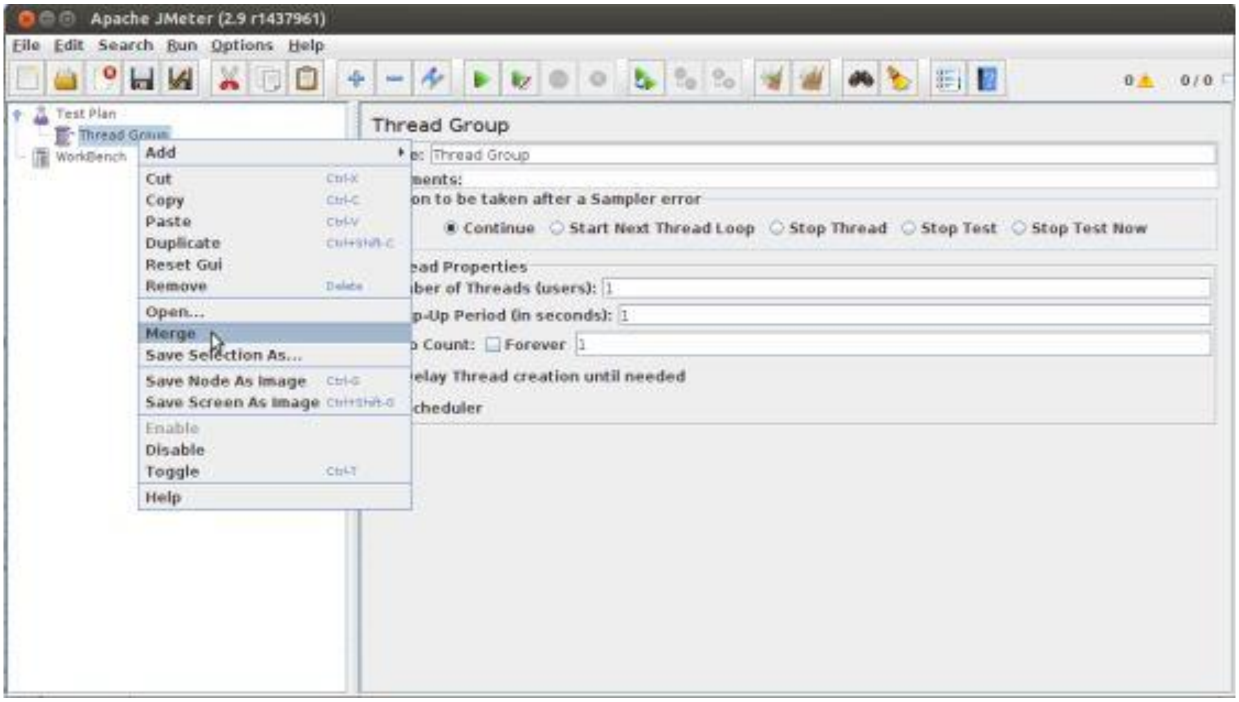

Apache JMeter (2.9 r1437961)
File Edit Search Run Options Help
0 0 / 0
Test Plan
Thread Group
WorkBench
Add
Cut Ctrl-X
Copy Ctrl-C
Paste Ctrl-V
Duplicate Ctrl+Shift-C
Reset Gui
Remove Delete
Open...
Merge
Save Selection As...
Save Node As Image Ctrl-G
Save Screen As Image Ctrl+Shift-G
Enable
Disable
Toggle Ctrl-T
Help
Thread Group
Thread Group
ents:
n to be taken after a Sampler error
● Continue ○ Start Next Thread Loop ○ Stop Thread ○ Stop Test ○ Stop Test Now
ad Properties
er of Threads (users): 1
-Up Period (in seconds): 1
Count: □ Forever 1
lay Thread creation until needed
heduler

Step 3: Load and Save the Elements

To load an element from file:

- Right-click on the existing tree element to which you want to add the loaded element.
-
- Select Merge.
-
- Choose the file where you saved the elements.
-
- JMeter will merge the elements into the tree.

By default, JMeter does not save the element, you need to explicitly save it.


Apache JMeter (2.9 r1437961)
File Edit Search Run Options Help
0 0 / 0
Test Plan
Thread Group
WorkBench
Add
Cut Ctrl-X
Copy Ctrl-C
Paste Ctrl-V
Duplicate Ctrl+Shift-C
Reset Gui
Remove Delete
Open...
Merge
Save Selection As...
Save Node As Image Ctrl-G
Save Screen As Image Ctrl+Shift-G
Enable
Disable
Toggle Ctrl-T
Help
Thread Group
e: Thread Group
ments:
on to be taken after a Sampler error
● Continue ○ Start Next Thread Loop ○ Stop Thread ○ Stop Test ○ Stop Test Now
ead Properties
ber of Threads (users): 1
p-Up Period (in seconds): 1
Count: □ Forever 1
elay Thread creation until needed
cheduler

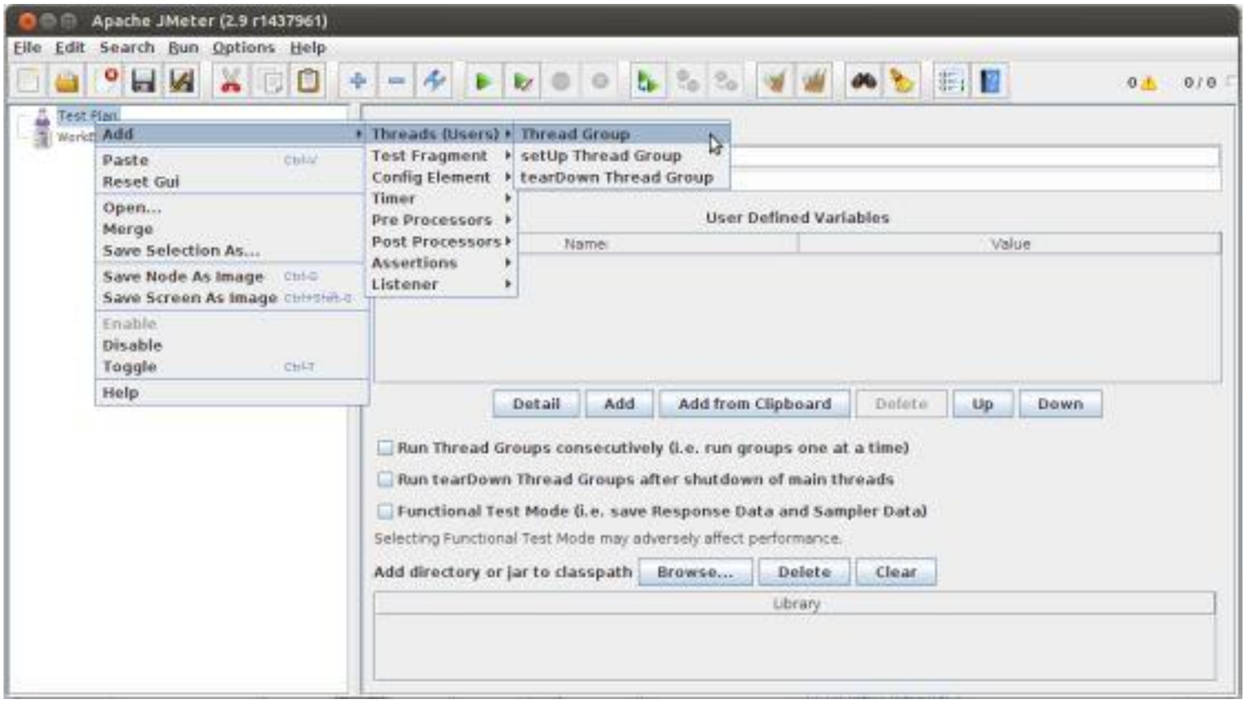
To save a tree element:

- Right-click on the element.
- Choose the *Save Selection As ...* option.

JMeter will save the element selected, plus all the child elements beneath it. By default, JMeter doesn't save the elements, you need to explicitly save it as mentioned earlier.

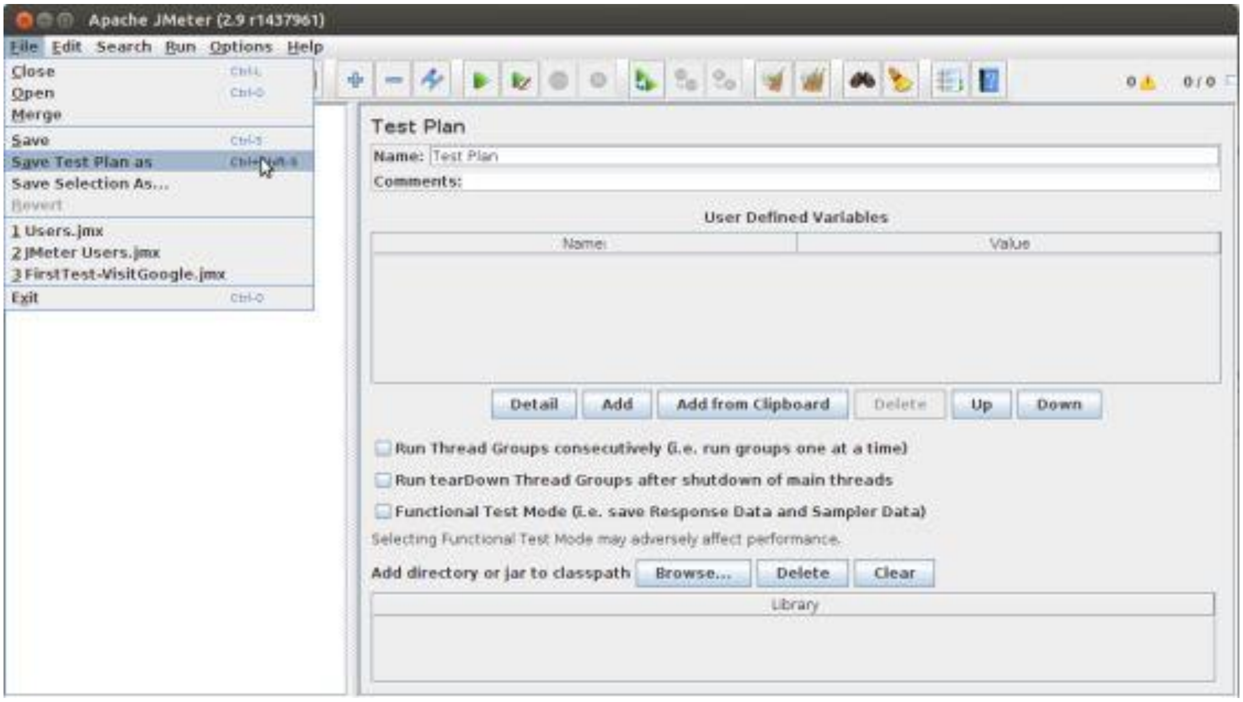
Step 4: Configure the Tree Elements

Any element in the Test Plan can be configured using the controls present in JMeter's right-hand side frame. These controls allow you to configure the behavior of that particular test element. For example, the Thread Group can be configured for a number of users, ramp up periods, etc., as shown below:


Apache JMeter (2.9 r1437961)
File Edit Search Run Options Help
Test Plan
WorkBench
Add
Paste Ctrl-V
Reset Gui
Open...
Merge
Save Selection As...
Save Node As Image Ctrl-G
Save Screen As Image Ctrl+Shift-G
Enable
Disable
Toggle Ctrl-T
Help
Threads (Users)
Test Fragment
Config Element
Timer
Pre Processors
Post Processors
Assertions
Listener
Thread Group
setUp Thread Group
tearDown Thread Group
User Defined Variables
Name:
Value
Detail
Add
Add from Clipboard
Delete
Up
Down
□ Run Thread Groups consecutively (i.e. run groups one at a time)
□ Run tearDown Thread Groups after shutdown of main threads
□ Functional Test Mode (i.e. save Response Data and Sampler Data)
Selecting Functional Test Mode may adversely affect performance.
Add directory or jar to classpath
Browse...
Delete
Clear
Library

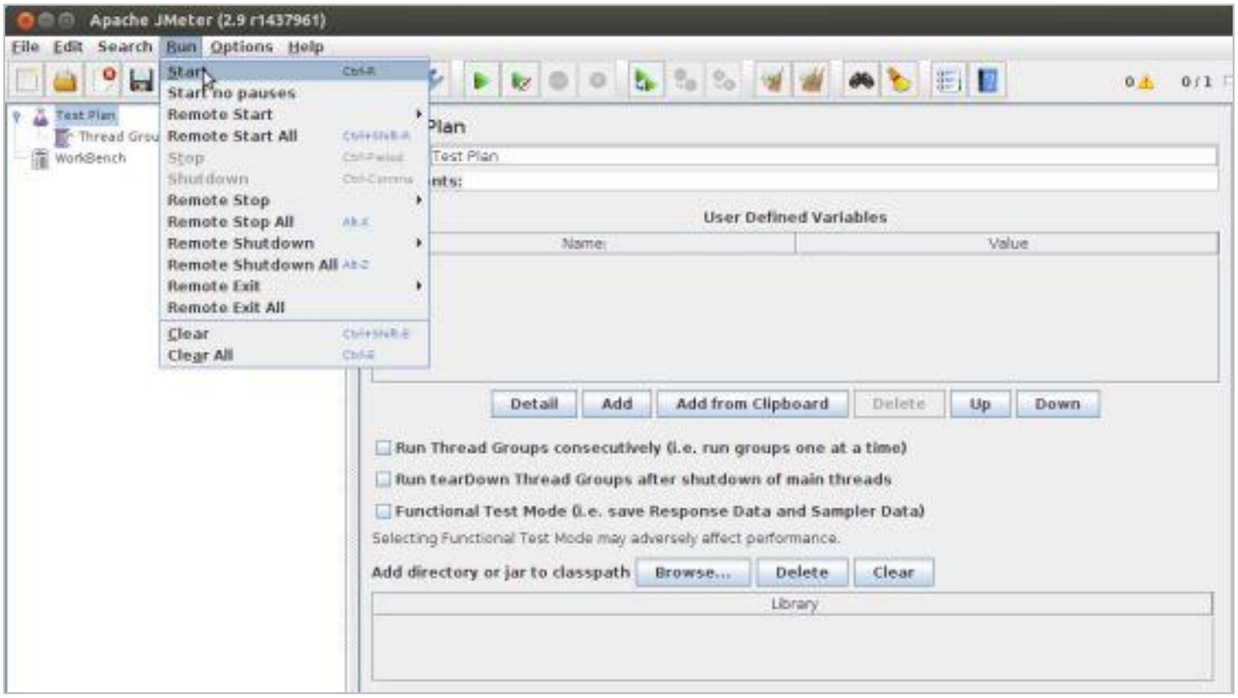
Step 5: Save the Test Plan

You can save an entire Test Plan by using either **Save** or **"Save Test Plan As ..."** from the File menu.


Apache JMeter (2.9 r1437961)
File Edit Search Run Options Help
Close Ctrl-L
Open Ctrl-O
Merge
Save Ctrl-S
Save Test Plan as Ctrl+Shift-S
Save Selection As...
Revert
1 Users.jmx
2 JMeter Users.jmx
3 FirstTest-VisitGoogle.jmx
Exit Ctrl-Q
Test Plan
Name: Test Plan
Comments:
User Defined Variables
Name:
Value
Detail
Add
Add from Clipboard
Delete
Up
Down
□ Run Thread Groups consecutively (i.e. run groups one at a time)
□ Run tearDown Thread Groups after shutdown of main threads
□ Functional Test Mode (i.e. save Response Data and Sampler Data)
Selecting Functional Test Mode may adversely affect performance.
Add directory or jar to classpath
Browse...
Delete
Clear
Library

Step 6: Run the Test Plan

You can run the Test Plan by clicking **Start** (Control + r) from the **Run** menu item. When JMeter starts running, it shows a small green box at the right-hand end of the section just under the menubar.


Apache JMeter (2.9 r1437961)
File Edit Search Run Options Help
Start Ctrl-R
Start no pauses
Remote Start
Remote Start All Ctrl+Shift-R
Stop Ctrl-Period
Shutdown Ctrl-Comma
Remote Stop
Remote Stop All Alt-X
Remote Shutdown
Remote Shutdown All Alt-Z
Remote Exit
Remote Exit All
Clear Ctrl+Shift-E
Clear All Ctrl-E
0 ⚠ 0 / 1
Test Plan
Thread Grou
WorkBench
Plan
Test Plan
nts:
User Defined Variables
Name:
Value
Detail
Add
Add from Clipboard
Delete
Up
Down
□ Run Thread Groups consecutively (i.e. run groups one at a time)
□ Run tearDown Thread Groups after shutdown of main threads
□ Functional Test Mode (i.e. save Response Data and Sampler Data)
Selecting Functional Test Mode may adversely affect performance.
Add directory or jar to classpath
Browse...
Delete
Clear
Library

The numbers to the left of the green box are the number of active threads / total number of threads. These only apply to a locally run test; they do not include any threads started on remote systems when using client-server mode.

Step 7: Stop the Test Plan

You can stop your test in two ways:

- Using **Stop** (Control + '.'). It stops the threads immediately if possible.
- Using **Shutdown** (Control + ','). It requests the threads to stop at the end of any current work.