**Introduction to JMeter**

Today, in the digital era, where websites and their users are increasing drastically, **seamless user experience** is a must-have for organizations because **unnecessary delays in the response** of the website or application distract the user's attention.

Now we have greater expectations of the software we use than ever before. This is the number one reason why performance testing has become so important.

Research suggests that just a 1 second delay in page load time results in 7% fewer conversions, 11% fewer page views, and a 16% decrease in customer satisfaction. And this translates to real dollars – if your site earns $100,000 per day, you’re losing $2.5 million every year due to this 1 second.

Therefore, organizations prefer doing performance and load tests on application before launching them or after every major release.

***What is performance testing?***

***Performance testing is a non-functional*** type of testing which is used to evaluate ***application under test*** (*AUT*) performance in terms of its responsiveness as well as the behaviour of AUT under the various workload.

An application is performant if it lets the end-user carry out a given task without undue delay or irritation.

An application which is performing well always have a **good response time**. So, responsiveness of an application plays a very important role just like it's user interface and functionality because nowadays users are very demanding and want everything quick.

**A slow website results into a bad user experience and have negative financial impact.** Even delay of second, for longer period of time, may result into huge revenue loss. Thus, performance test of websites comes into picture. To performance test a website there are tools available, for example: ***JMeter, LoadRunner, WebLoad, LoadView*** and many more.

***Importance of Performance Testing***

* *Helps in evaluating potential bottlenecks of an AUT*.
* *Slowness of an application or web service can be evaluated under heavy load.*
* *We can find out how many parallel users an AUT can handle.*
* *Helps in finding out impact of changes in each release in terms of performance*.

***Note: An application or software system which is when tested is called AUT or Application Under Test***..

***What is JMeter and Its Brief History?***

[***Apache JMeter***](https://jmeter.apache.org/) **is an open-source, pure Java platform software** which is designed to **load test functional behaviour and measure performance**.

Initially, JMeter was introduced for load and performance test web applications, but later on its scope has widened and can perform load and performance test on Web Pages, Web Applications and static or dynamic resources Like ***Database, Rest Webservices, LDAP, Java Objects*** and more.

**Features of JMeter**

***Key features of JMeter includes:***

* ***License****: Since* ***JMeter is open source****, it is free and easily available*.
* ***Graphical User Interface****: Simple, user friendly and easy to learn as compared to other performance testing tools*
* ***Server/ Protocol Suppor****t: JMeter has ability to load and performance test different applications/server/protocols. A few protocols inclucdes* ***HTTP, HTTPS, FTP, SOAP/REST, Database via JDBS, LDAP, JMS, SMTP(S), POP(3) and IMAP(S), Native Commands/ Shell Scripts and TCP*.**
* ***Platform****: JMeter is pure java software. Therefore, it is platform-independent and supports all environment.*
* ***Simulation****: Simulate multiple users by using virtual users or unique users in order to generate heavy traffic on web server or services*.
* ***Supports Distributed Testing****: It has master slave for distributed testing where master will distribute tests among all slaves and slaves will execute scripts against your server.*
* ***Test Result Visualisation****: Test result can b view in different formats like Graph, Table, Tree, and Report, etc.*
* ***Reporting:****By default, JMeter provides XML and CVS Report Formats only. We can use Jmeter and ANT together to obtain HTML report as per the requirement.*
* ***Testing Types****: Apart from just Performance, Load, Stress testing Jmeter works well for Functional, Regression and Soak/Endurance testing too*.
* ***Record and Playback:****Record user scenario/action in Firefox Browser and play scripts*
* ***Framework****: Multi-threading framework allows concurrent and simultaneous sampling of different functions by many or separate thread groups*.
* ***Installation****: No Complex installation required – Just run JMeter.bat on windows / run Jmeter.sh on Linux*
* ***Knowledge****: Jmeter does not require extensive programming knowledge. Only prior knowledge Java language is preferred.*

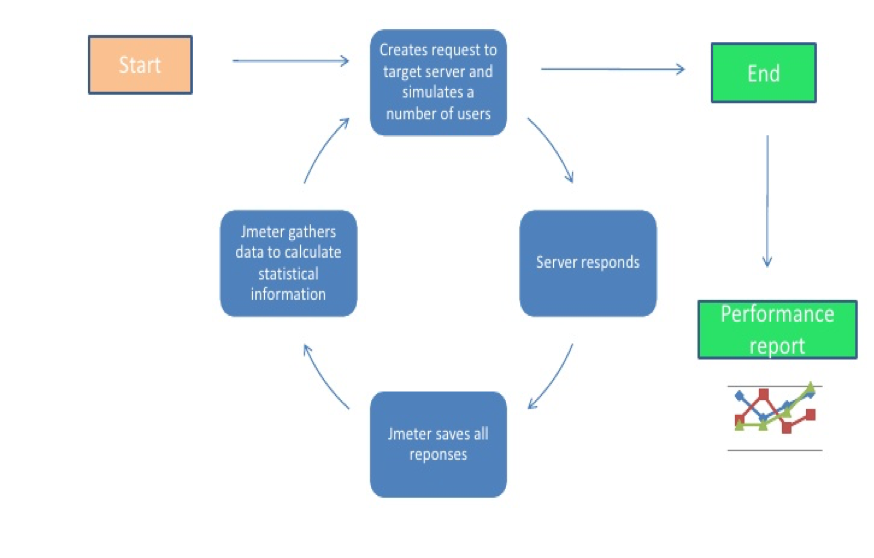
**How JMeter Works**

**JMeter simulates a number of users sending request to the Application under Test**.  As soon as JMeter simulates requests, server responds and Jmeter starts collecting data. Jmeter saves all the responses and based on the server response it return statistics. These statistics shows performance of the AUT in the form of various formats as per the requirements.

Thus, with the help of JMeter we can simulate load on server, network or objects which are coming from different machines to implement real world scenario.

***JMeter Workflow***

When we start load or perform test of an application, JMeter creates requests to target server and simulates number of users sending requests to the target server. As soon as server starts responding to the requests, JMeter starts saving all the responses. On the basis of data/response JMeter gathers data to calculate statistical information. Finally, using this statistical information JMeter prepares a report which tells about performance of the AUT.

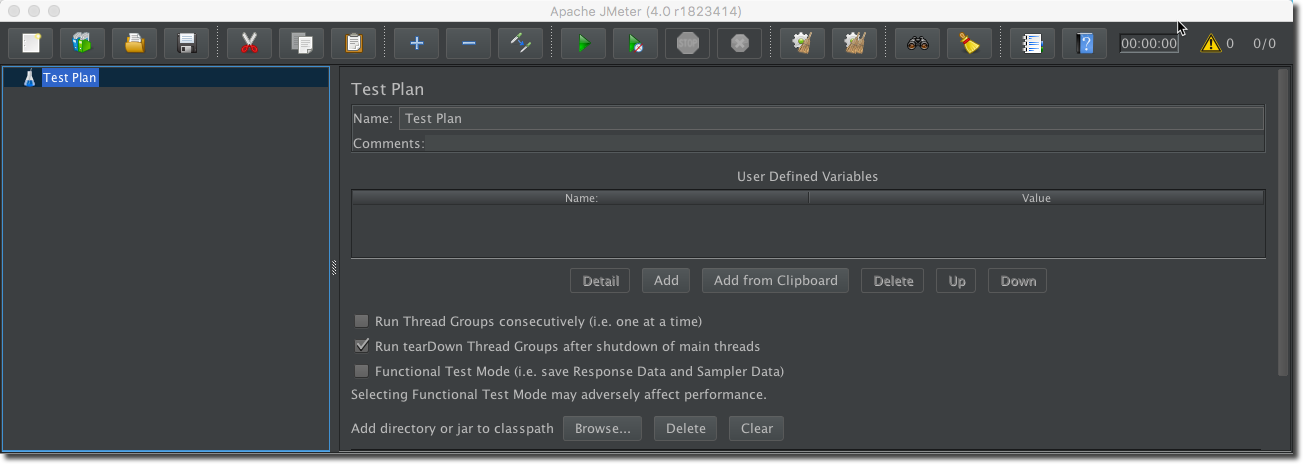


So, this was a brief about what is JMeter and how is it useful for us. In the upcoming section we will be installing this open source, user friendly software and slowly we will try to get hands on this performance testing tool.

How to launch Jmeter

To launch ***JMeter on Windows just double click the jmeter.bat*** file or go to ***command prompt*** and type ***<file path>/apache-jmeter-4.0/bin/jmeter.bat*** and wait for a few seconds JMeter GUI will be launched.

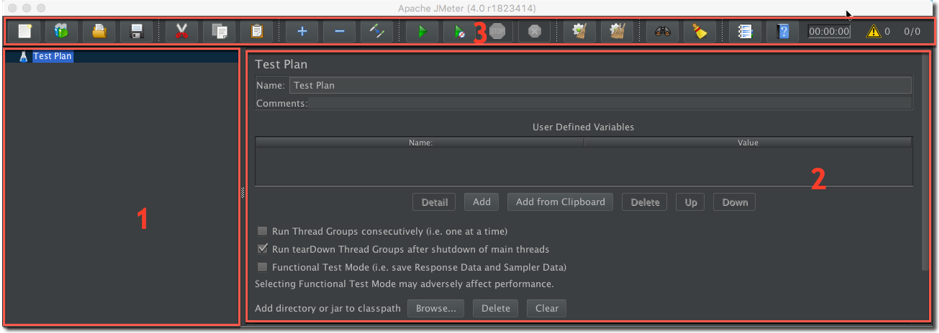
# Getting Familiar with JMeter GUI



JMeter is primarily divided into three major parts:

* ***Left Pane***: Left pane is the place where the test you want to execute resides.
* ***Configuration Window***: In this window we set the configuration and control the behaviour of the tests that we want to execute.
* ***Menu Bar***: It is an intuitive menu bar from where you may perform all the functions.

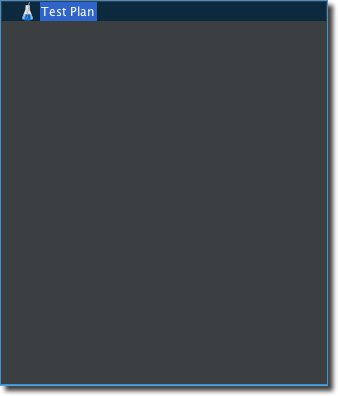
The ***options menu*** has been removed from the GUI in JMeter 4.0. So, the user may perform nearly all the functions using the buttons present in the ***Menu Bar***.



## Left Pane

Left pane in JMeter consists of a node known as ***Test Plan***. As the name suggests, Test Plan can be thought of as a container that consists of test scenarios as well as test data.

Similarly, JMeter Test Plan also can be thought of as a container that consists of series of steps that will execute when we run a test plan. Test plan can consist of one or more elements like ***Thread Groups, Logic Controllers, Configuration Elements, Samplers, Timers, Listeners and Assertions***.

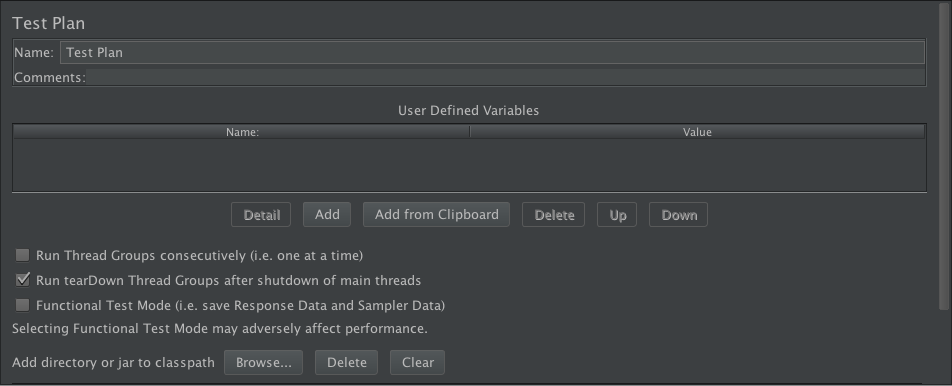


***Note*: Only one test plan is executed at a time in JMeter.**

## Configuration Pane

You can **configure** your Test Plan and its element on this window. A Test Plan’s name, user defined variables and its properties can be configured here. Configuring a Test Plan helps you to control its properties and the behaviour in which you would like to run your test Plan as per your requirement.

By default the below screen is visible when we launch JMeter.

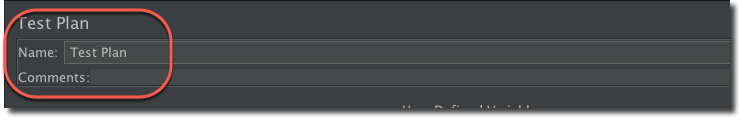


The above screen consists of three major parts:

* ***Name***
* ***User Defined Variables***
* ***Test Plan Properties.***

#### **Name**

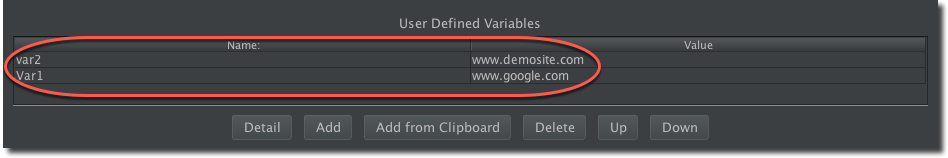
This shows the name of your test plan. You may change the name of your test plan here. We will be discussing how to create, name and save a Test Plan in our next tutorial.



#### **User- Defined Variables**

User-defined Variables are nothing but ***name-value pairs***.

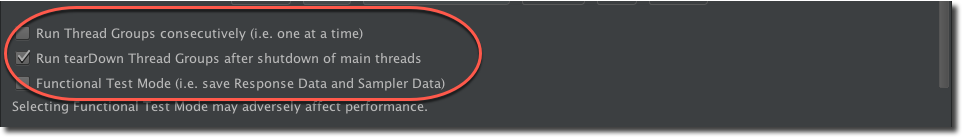
You can add a variable by clicking on ***Add button***. You can add multiple name-value pairs either by clicking ***Add button*** every time or by using ***Add from Clipboard*** button. The following screen shows the added variables along with their values.



#### **Test Plan Properties**

There are three major configuration properties of a Test Plan which you may use to control test plan’s behaviour as per requirement. The three properties are:

* ***Run Thread Groups Consecutively(i.e. one at a time)***
* ***Run tearDown Thread Groups after shutdown of main threads***
* ***Functional Test Mode(i.e. save Response Data and Sampler Data)***



***Run Thread Groups Consecutively (i.e. one at a time)***

A Test Plan can have one or more Thread Group. It should have at least one Thread Group (which is the minimum requirement) but can have more than one too. If a Test Plan has more than one thread group then they will execute one after the other if this checkbox is checked. If this checkbox remains unchecked then all the thread groups will execute in parallel.

***Run teardown Thread Groups after shutdown of main threads***

TearDown threads execute after the test has finished executing its regular thread group. If this checkbox is checked, then this thread will run after the test execution. The tearDown feature is generally used for reporting or cleaning operations. ***Example***, If you want your logs to clean automatically after the execution of a Test Plan or want your reports to be in particular format then you may use this property of the Test Plan.

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

If this checkbox is selected, then sampler requests and response data are saved in the listeners. Do not worry about the listeners and other technical terms here. We will discuss everything in the later tutorial and this tutorial is just to get you familiar with the GUI of JMeter. This checkbox allows you to verify that the test is working as expected.

***Note: A Thread Group is a child element of a Test Plan. Each Thread Group represents a use-case. (We will discuss about Thread Group in our upcoming tutorials)***

## Menu Bar

This is the upper most bar present on the JMeter GUI. It has many buttons which helps us to perform various function just by clicking the respective button.

We can perform many operations using this above menu bar but discussing all of them here is not necessary. Hence we will only be discussing the most important one. If any other function comes later in the course, we will detail it there.

MenubAr

The below images shows the mostly used icons followed by their working.

FirstMenu

* ***New***: Using New menu item you can create a new Test Plan.
* ***Open***: To open an existing Test Plan you may use open button present in the menu bar.
* ***Save***: To save the Test plan or its elements you can simply click this button. As you will click this button a prompt appears

SecondMenu

* ***Start***:  Start icon used to execute the test. After creating a test plan and adding elements to it you may simply click this button to start the execution of your test.
* ***Start with no pauses***: Like Start we can use Start with no pauses to execute the test. Difference between both is that in case of Start No Pause timer configured in the thread group are skipped and the thread will run without any manual pauses.
* ***Stop***: As the name suggests, you can use Stop to abruptly stop the test that is running.
* ***Shutdown***: Shutdown also stops the execution but gracefully. When you shut down the running test it does not stop it immediately but allow the running threads to wind down.

ThirdMenu

* ***Clear All***: You can clear the log window by clicking Clear All button.
* ***Elapsed time of current running test***: Shows the time taken to execute the test.
* ***Show Number of Errors in log***: On clicking this menu option you will get the logs at the bottom of the configuration window.
* ***Running thread/total number of threads***:  this option tells you how many threads are currently running out of the total number of threads.

Hence, in this tutorial we learned about ***Getting Familiar with JMeter GUI*** and got to know in detail what is a test plan and what are the properties of a test plan. In the next tutorial we will be learning  about how to perform some of the basic operations in JMeter.