

Software Testing Tools:

WinRunner

WinRunner is the most used Automated S/W Testing tool.

Main features of WinRunner are

* → Developed by Mercury Interactive.

* → functionality testing tool.

* → supports C/S and web technologies such as (VB, VC++, Delphi, C/C++ (ERP))

* → To support .NET, XML, SAP, Peoplesoft, Oracle applications, Multimedia we can use QTP.

* → WinRunner runs on windows only.

* → XRunner runs only on UNIX & Linux.

* → Tool developed in C on VC++ environment.

* → To automate our manual test WinRunner used TSL (Test Script language like C).

* → The main testing process in WinRunner is

1) Learning

Recognition of objects and windows in our application by WinRunner is called "Learning".

2) Recording

WinRunner records over manual business logic operation in test.

3) Edit script

depends on corresponding manual test, test

engineer inserts check points given to that record script

4) Run script

During test script evaluation, execution, WinRunner compare tester given expected values and application actual values and return result.

5) Analyze Results

Tester analyzes the tool given results to concentrate on defect tracking if required; how does the WinRunner Tool work? WinRunner is a functional testing tool to work on compilation of tests that are used for collaborating with the HP Quick Test Professional and to be used as a supporting element for the quality assurance process, as a part of test phase in the software development life cycle. This testing process is applied as a part of process into product excellence.

*→ The testing process here goes by capturing the functional requirement / test requirement, by verifying the actual results against the expected results and by replaying the user operations / functional activities performed on the software product.

Advantages:

- * → enable rapid testing
- * → provides consistency
- * → Reusability of tests
- * → Customizable for future changes

Disadvantages

- * → Doesn't apply for stress/parallel load or scalability + testing of legacy systems if fail
- * → Doesn't support native programming
- * → Tester should have programming knowledge/experience
- * → It doesn't support multimedia systems

Selenium

selenium automation uses the selenium toolbox to run tests across different browser instances

- * → selenium grid lets you test across multiple instances and machines at once. There are a lot of types of testing it can be used for:
- * system testing
- * end-to-end testing
- * compatibility testing
- * regression testing
- * integration testing
- * performance testing

Essentially, any test you can think of that can be run in a browser can be run in selenium if you have the knowledge and training to setup.

while Selenium testing is likely to have a place in developer's toolkits for some time to come, there are a number of things that make it less practical for today's complex testing requirements.

- * → It's complex with a steep learning curve.
- * → It takes a lot of manual coding to build the testing solution you want.
- * → It takes a lot of training to get the most out of it.
- * → No built-in reporting capabilities.
- * → No support apart from user forums.

Components of selenium:

* Selenium IDE

* Selenium WebDriver

* Selenium Grid

How does Selenium work?

Selenium WebDriver works by emulating the actions of a user. When you write a Selenium test, you specify the action you want the user to take and Selenium will automatically execute those actions on the browser.

For instance, if you need to test a login functionality, you would write a Selenium test to enter the username and password into the appropriate fields and click on the login button. Selenium would automatically execute those

actions on the browser and moreover any

Advantages:

- * → selenium is open source, free ware, and portable tool.
- * → selenium supports variety of languages that include java, perl, python, c++, ruby, java script, groovy and VB script etc.
- * → selenium supports many operating system like windows, macintosh, Linux, unix etc.
- * → selenium supports many browsers like chrome, internet explorer, firefox, safari, opera etc.

Disadvantages:

- * → It is difficult to test image based application
- * → selenium doesn't support build in add ins support.
- * → selenium script execution time is bit high
- * → selenium does not support file upload facility
- * → selenium partially supports four dialog boxes

Apache JMeter

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

stefano mazzocchi of the apache s/w foundation was the original developer of JMeter.

He wrote it primarily to test the performance of Apache Jserv.

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 graphic API to implement its graphical user interface.

* → The protocols supported by JMeter are:

* → Web - HTTP, HTTPS sites

* → Web services - SOAP, XML-RPC

* → Database via JDBC drivers

* → Directory - LDAP

* → messaging oriented service via JMS

* → service - POP3, IMAP, SMTP

* → FTP service

How JMeter works:

* → JMeter simulates requests to a target server, and returns statistics that show the performance/ functionality of the target server/application via tables.

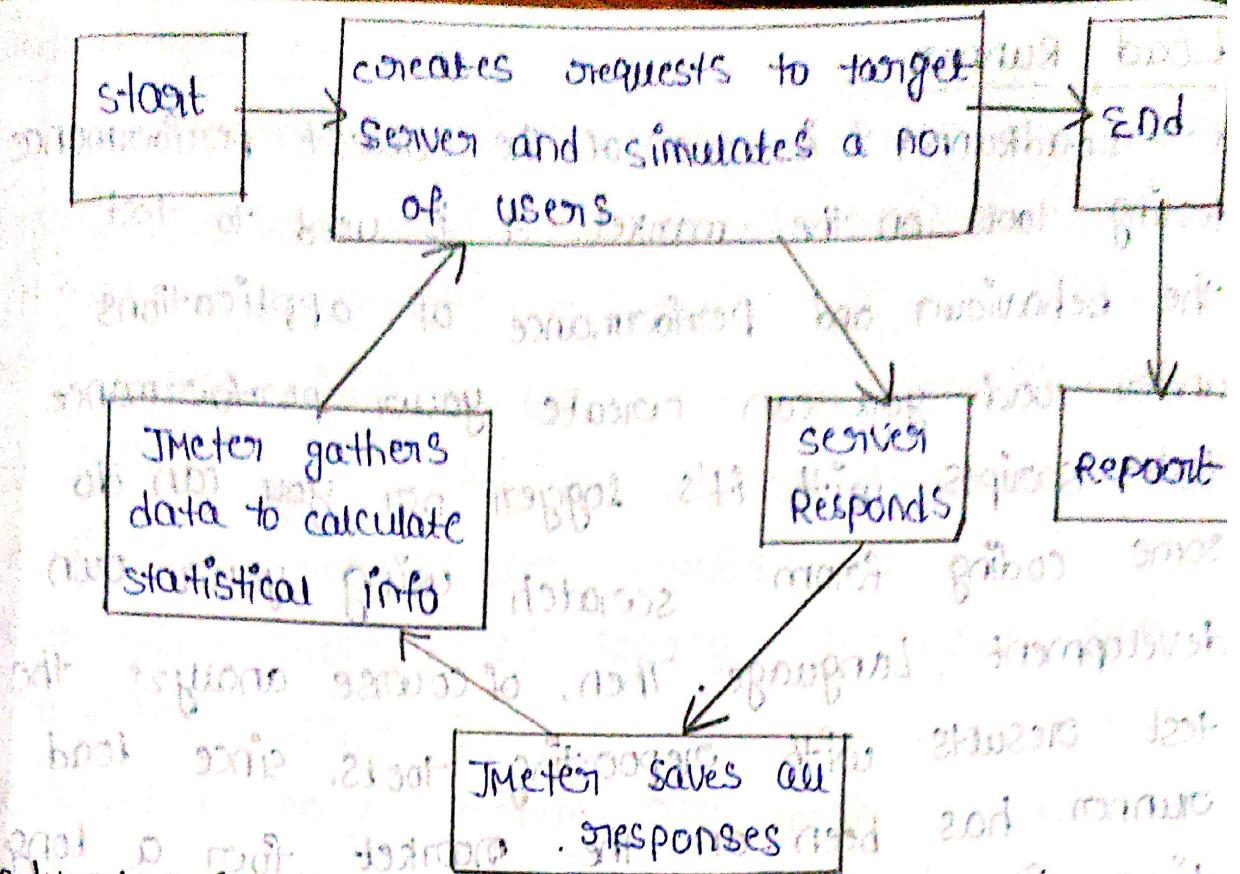
Take a look at the following figure that depicts how JMeter behaves when it works:

1. Application is started and as fast as possible with regards to mission critical

2. It needs to establish connection with each of the static sites

3. It needs to establish connection with each of the dynamic sites

4. It needs to establish connection with each of the static sites



Advantages of JMeter:

- * → Provides integration with Jenkins and reporting.
- * → Easy installation on any operating system.
- * → Test IDE allows test recording from native applications.
- * → Allows API testing, Database testing and MQ testing with ease.
- * → Key features like the thread group, helps to see whether performance is good or bad.

Disadvantages:

- * → Automation is difficult with JMeter.
- * → JMeter output reports are difficult to understand without training.
- * → It doesn't support testing of Javascript and AJAX requests.
- * → It's difficult to maintain.

Load Runner

* LoadRunner is one of the oldest performance testing tools on the market. It is used to test the behaviour and performance of applications under load. You can create your performance test scripts with its logger or you can do some coding from scratch using your own development language. Then, of course analyze the test results with reporting tools. Since LoadRunner has been on the market for a long time, it supports many different technologies.

How does LoadRunner work? LoadRunner works on the concept of recording and replaying user activities and generating designed load on the server. It simply simulates the actions of the user in the real world and creates a virtual load helping to determine the performance of the software application or system. The main steps include:

- 1) Scripting / Recording: To record user action in a script

- 2) Test Execution: Replaying the script along with the virtual payload to simulate the real world situation in the test environment.

- 3) Result Analysis: To provide accurate result in terms of load carrying capacity and responsiveness of the application.

Advantages of LoadRunner:

- *→ LoadRunner can accurately detect system level, end-user and code based bottlenecks in code and interface fixtures.
- *→ Helps minimize the cost of application downtime by identifying the root cause of application performance issues.
- *→ Allow testing of legacy software with updated technologies.
- *→ Allow easy - mobile app testing.
- *→ It makes optimum use of load generator and gives better, load test results.
- *→ optimum use of load builder farm

Disadvantages of Load Runner:

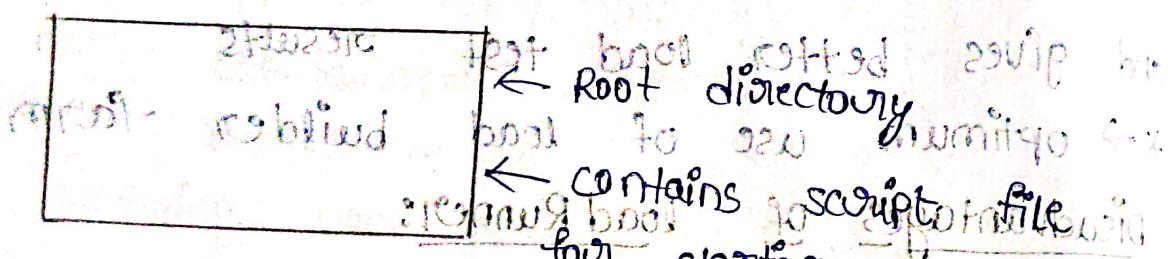
- *→ Much more expensive compared to other test tools available on the market.
- *→ Debugging features need fundamental improvements for better results.
- *→ It uses a lot of memory space and has major capability issues with other frameworks.
- *→ It has limited load generation capacity.

* Apache JMeter

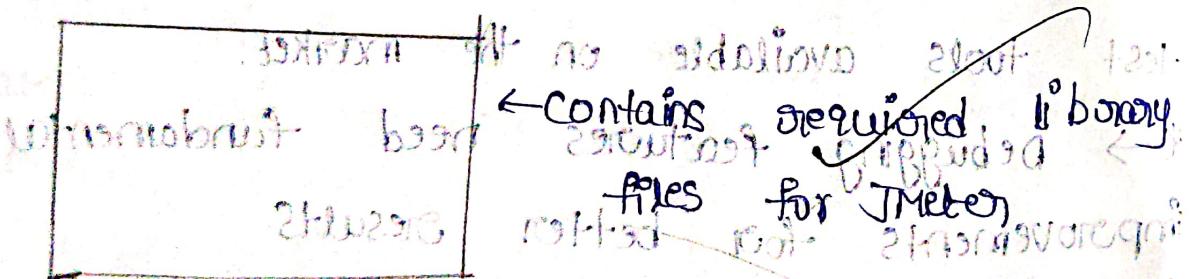
Installation

Installation of JMeter is extremely easy and simple. You simply unzip the zip/tar file into the directory where you want JMeter to be installed.

There is no tedious installation screens to deal with. Simply unzip and you are done! Once the unzipping is done installation directory structure should look like as figure below.



Path of JMeter : C:\Program Files\Apache Software Foundation\Apache JMeter 2.11\bin\jmeter.bat



Path of JMeter : C:\Program Files\Apache Software Foundation\Apache JMeter 2.11\bin\jmeter.bat

JMeter directory contains many files & directory.

* → /bin : contains JMeter script file for starting JMeter

* → /docs : JMeter documentation files

* → /extras : and related extra files

*→ /lib/: contains the required Java library

→ four JMeter

*→ /lib/ext/: contains the core jar files for JMeter

→ JMeter and the protocols

*→ /lib/junit/: JUnit library used four JMeter

*→ /printable-docs:

Launch JMeter no way shown course of book if you can start JMeter in 3 modes

→ GUI mode

→ Server mode as baseline and not make test it

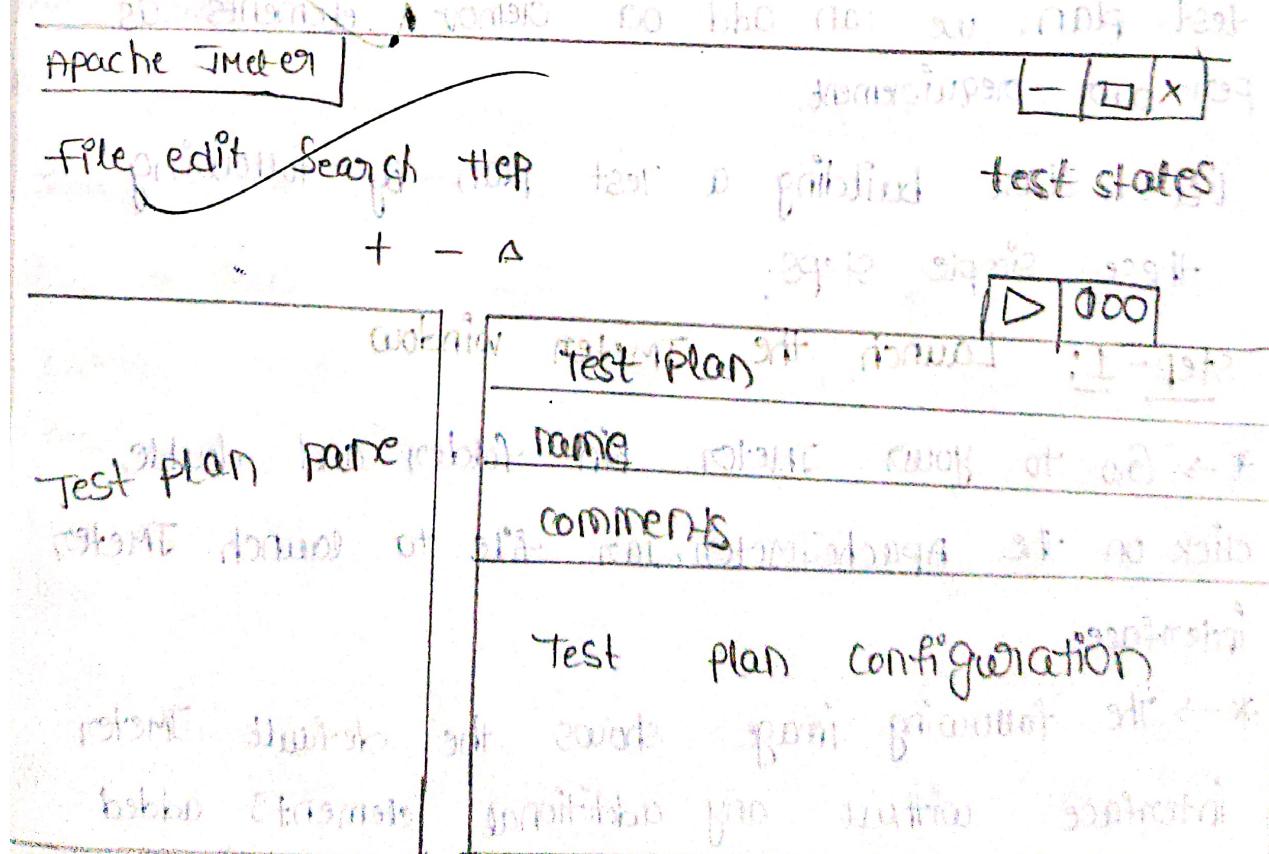
→ command Line mode, start gains not ignore

Start JMeter, in GUI mode as true choice test

If you are using windows, just run the file

/bin/jmeter.bat to start JMeter in GUI mode

as shown below



Start JMeter in server mode

Server mode is used for distributed testing. This testing works as a client-server model. In this model, JMeter runs on a server computer in server mode. On a client computer, JMeter runs in GUI mode.

To start the server mode you can run the bat file bin\Jmeter\server.bat as below:

Build JMeter test plan

A test plan can be visualized as a JMeter script for running tests. A test plan consists of test elements such as thread groups, logic controllers, sample generating controllers, listeners, timers, assertions and configuration elements.

There should be at least one thread group in every test plan. we can add or remove elements as per our requirement.

Let's start building a Test plan by following these simple steps:

Step-1: Launch the JMeter window

* → Go to your JMeter bin folder and double click on the ApacheJMeter.jar file to launch JMeter interface.

* → The following image shows the default JMeter interface without any additional elements added to it:

*→ The Jmeter interface contains a test plan node where the test plan is kept.

*→ The test plan node contains Name of the test plan and user defined variables.

*→ User defined variables provides flexibility when you have to repeat any value in several parts of the test plan.

Step-2: Add/ Remove test plan elements

*→ Once you have created a test plan for Jmeter, the next step is to add and remove elements to Jmeter test plan.

*→ Select the test plan node and right click on the selected item. From the dropdown menu, click "Add".

*→ Mouse hover on "Add" option, then elements list will be displayed.

*→ Mouse hover on designed list and select the desired option by clicking. Right click to refresh.

*→ The following image shows how to add a thread group element in a test plan.

*→ To remove test element, select the desired test element. Right click on the test element and choose the "Remove" option. The following image shows how to remove a thread group element in a test plan.

Step-3: Load and save test plan elements

*→ To load elements to Jmeter test plan, select and right click on any tree element

on which you want to add the loaded elements.

* → select "merge" option.

* → choose the .jmx file where you save the elements.

* → Elements will be merged into the JMeter test plan.

* → To save those elements, right click on the element.

* → choose "Save Selection As" option. Save file on desired location.

Step-4: configuring the three elements

Elements in the test plan can be configured by using controls present on JMeter's right hand side for me. These controls allow you to configure the behaviour of the selected element. For example,

a thread group can be configured by its name, Number of threads (the number of users you are testing).

Ramp-up time (how much time you want to allow the Thread group to go from 0 to 3 users).

Loop Count (how many times the test should be looped).

Step-5: save JMeter test plan.

Till now we are done with creating a test plan, adding an element and configuring a thread.

Now, you can save the entire test plan by choosing the "Save As" from file menu.

As " " from file menu.

Types of threads (users)

open model Thread group

Thread group

Setup Thread Group

teardown Thread group

Types of Listener:

View Results Tree

summary Report

Aggregate Report

Backend Listener

Types of Assertions:

Response Assertion

JSON Assertion

Size

JSR223

Xpath2

Experiment - 1

Creating a test plan

Step-1: Start JMeter

Step-2: Create a test plan

Step-3: Create a thread group (i.e., right click on test plan → add → Thread → Thread Group)

Step-4: Add a sampler i.e., HTTP request

www.gmail.com

Path: /

Step-5: Add listeners

summary - view results tree

Step-6: Save test plan with an extension of ".jmx"

Step-7: Run the test plan

Experiment - 02:

Timers

JMeters sends request without applying any delay b/w each other i.e., b/w each sampler or request.

If you perform local testing (00) stress testing on your server without any delay it will be overloaded then it will not be able to give realistic results & failed to stimulate

real world user traffic experience
the timer element can be added in a test plan
to apply wait b/wn each Sampler request
*→ Timer element i) constant timer
ii) uniform random timer

Step-1: create a thread group

Testplan → add → thread → Thread Group

Step-2: create a Sampler i.e., HTTP request

Testplan → add → sampler → HTTP request

server : www.gmail.com

path : /

Create another Sampler HTTP request 1

Create another Sampler & rename it as

HTTP request 2

Testplan → Add → sampler → HTTP request 1

✓ server : www.google.com

Path : /

Step-3: Create a Listener

Testplan → Add → Listener → summary

→ view results tree

Uniform Random Timer:

⇒ In Uniform Random timer we find two options

- 1) Random delay maximum set to 3000
- 2) constant delay offset which is set to -3000

uniform Random Timer

In uniform Random timer we find 2 options

delay maximum set to 3000
offset - 3000

WSS
offset