Jmeter

The **Apache JMeterTM** is pure Java **open source** software, which was first developed by Stefano Mazzocchi of the Apache Software Foundation, designed to load test functional behavior and measure performance. You can use JMeter to analyze and measure the performance of web applications or a variety of services. Performance Testing means testing a web application against heavy load, multiple and concurrent user traffic. JMeter originally is used for testing Web Application or FTP application. Nowadays, it is used for a functional test, database server test etc.

JMeter Advantages

- **Open source license**: JMeter is totally free, allows developer use the source code for the development
- **Friendly GUI**: JMeter is extremely easy to use and doesn't take time to get familiar with it
- **Platform independent**: JMeter is 100% pure Java desktop application. So it can run on multiple platforms
- **Full multithreading framework**. JMeter allows concurrent and simultaneous sampling of different functions by a separate thread group
- **Visualize Test Result:** Test result can be displayed in a different format such as chart, table, tree and log file
- **Easy installation**: You just copy and run the *.bat file to run JMeter. No installation needed.
- **Highly Extensible**: You can write your own tests. JMeter also supports visualization plugins allow you to extend your testing
- **Multiple testing strategy**: JMeter supports many testing strategies such as Load Testing, Distributed Testing, and Functional Testing.
- **Simulation**: JMeter can simulate multiple users with concurrent threads, create a heavy load against web application under test
- **Support multi-protocol**: JMeter does not only support web application testing but also evaluates database server performance. All basic protocols such as HTTP, JDBC, LDAP, SOAP, JMS, and FTP are supported by JMeter

● **Record & Playback** - **Record** the user activity on the browser and simulate them in a web application using JMeter

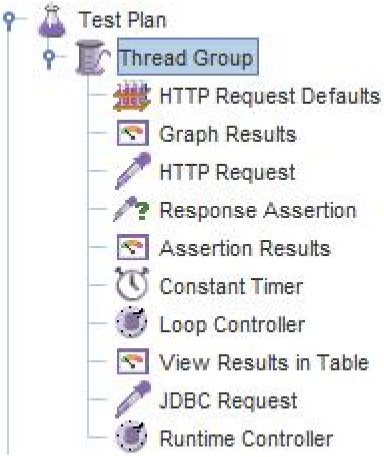
● **Script Test**: Jmeter can be integrated with Bean Shell & Selenium for automated testing.

What is a Test Plan?

Test Plan is where you add elements required for your JMeter Test.

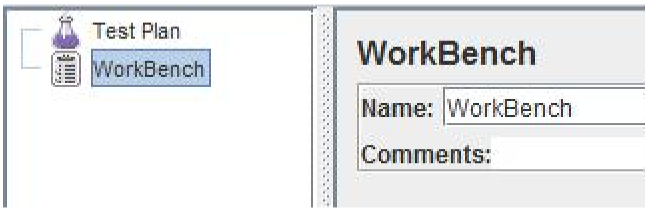
It stores all the elements (like ThreadGroup, Timers etc) and their corresponding settings required to run your desired Tests.

The following figure shows an example of Test Plan



What is WorkBench?

The WorkBench simply provides a place to store test elements **temporarily**. WorkBench has no relation with Test Plan. JMeter will **not save** the contents of the WorkBench. It only saves the contents of the Test Plan branch"WorkBench" will be used in our tutorial HTTP Proxy Server Recording Test

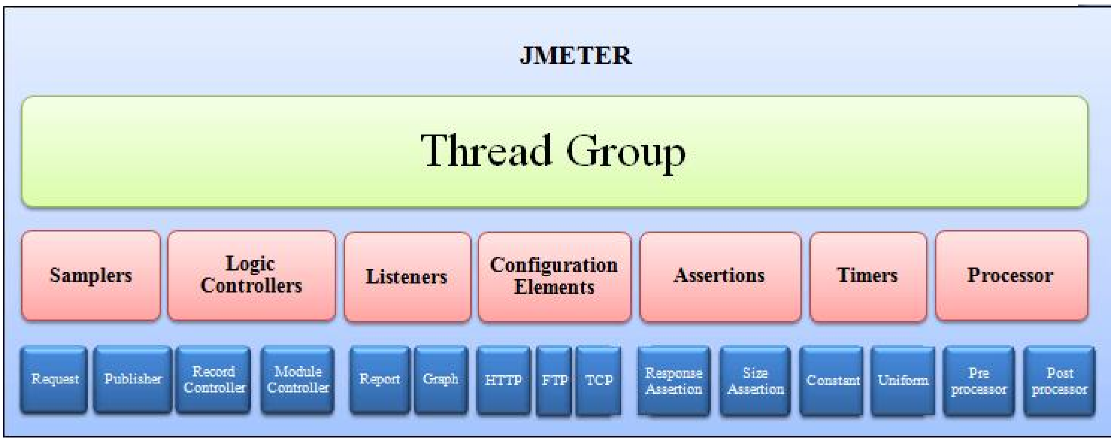


Name: WorkBench

Comments:

What is Element in JMeter?

The different components of JMeter are called Elements. Each Element is designed for a specific purpose.



Thread Group

As the name implies, thread group represents the group of threads JMeter will use during the test. Thread group elements are the beginning points of any test plan. The controls provided by a thread group allow you to:

- Set the number of threads.
- Set the ramp-up period.
- Set the number of times to execute the test.

The Thread Group Control Panel includes:

- Itsname.
- 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).
- Scheduler checkbox (The checkbox at the bottom of the Thread Group panel is used to enable/disable extra fields in which you can enter the duration of test, the startup delay, the start and end times of the run).

Controllers

Controllers can be divided into two broad categories:

- Samplers
- Logical Controllers

Samplers

Samplers are the components which allow JMeter to send specific types of requests to a server.It simulates a user's request for a page to the target server.

Samplers are a must to add component to a test plan as only it can let JMeter know what type of request need to go to a server. Requests could be HTTP, HTTP(s), FTP, TCP, SMTP, SOAP etc.

Some of the most widely used Sampler in JMeter is-

- HTTP Request – Used to send HTTP/HTTPS requests to server. This is the most widely used sampler for testing Web-based applications.
- JDBC Request – Used to send SQL queries to a database server.
- SOAP/XML-RPC Request – Used to send SOAP requests to a SOAP web service.
- Test Action – This is a special type of Sampler, which doesn't send a request to the server instead it is used to introduce pauses in a test.
- FTP Request – Used to send file put and get requests to an FTP server.

Given below is the list of requests serviced by JMeter samplers:

- FTP Request
- HTTP Request (can be used for SOAP or REST Webservice also)
- JDBC Request
- Java object request
- JMS request
- JUnit Test request
- LDAP Request

- Mail request
- OS Process request
- TCP request

Logical Controllers

Logic Controllers help you to control the flow the order of processing of samplers in a thread. It can also change the order of requests coming from their child elements.

Following is the list of all Logic Controllers in JMeter:

- Runtime Controller
- IF Controller
- Transaction Controller
- Recording Controller
- Simple Controller
- While Controller
- Switch Controller
- ForEach Controller
- Module Controller
- Include Controller
- Loop Controller
- Once Only Controller
- Interleave Controller
- Random Controller
- Random Order Controller
- Throughput Controller

- Critical Section Controller

This is a newly added controller in JMeter. It ensures that the child elements of this controller are accessed by only one thread at a time.

- ForEach Controller

This controller is used to perform the requests in a loop, based on the values of a set of related variables.

- If Controller

Using If Controller, we can specify a condition whether the child element inside it will run or not based on a condition that should evaluate to true or false.

- Include Controller

The Include Controller can be used to provide modularity in JMeter. Using this controller we can add an external .jmx file(a test fragment) to our existing script by loading the jmx file in the Include Controler's control panel.

- Interleave Controller

Interleave Controller allow us to pick and execute a single child element out of multiple child in each loop iteration. For example- if we have three samplers added as child to an Interleave Controller than the in first iteration it will pick the first sampler request, in next iteration it will pick the next one and so on. Once the child elements are over, it again starts iteration from first element.

- Loop Controller

It allows to execute the operations specified as child elements in a loop with iteration value specified in its control panel.

- ❖ Module Controller

Using Module Controller, we can reuse a test fragment(e.g. a sampler) into our script again by selecting the module from the Module Controller's control panel.

- ❖ Once Only Controller

The once only controller is used in situation where we would like to perform an operation only once even if the operation is executed in a loop.

- ❖ Random Controller

It is similar to a random controller and picks a single child element in each iteration but unlike the Interleave controller it picks the child element randomly.

- ❖ Random Order Controller

The Random Order controller is used to execute each of its child element at most once in random order.

- ❖ Recording Controller

This controller acts as a placeholder where the scripts recorded using HTTP Proxy Server are recorded by default(for details on Record and Playback in JMeter check our tutorial).

- ❖ Runtime Controller

Runtime controller is used to limiting the time of execution of its child elements. For example, if we have specified the value of 'Runtime' as 100 seconds then the elements inside the Runtime controller will run for 100 seconds with as much iteration as possible.

- Simple Controller

This controller is just a placeholder for grouping and ordering the different elements of the test plan.

- Switch Controller

This controller is used to pick one element for processing out of its multiple child elements. The element is picked not in sequential order or random order instead it is based on a switch value defined in its control panel. The switch value can be a variable with value evaluated to its index(position) of the element or name of the element. For example, if in the first iteration the switch value is evaluated to ‘3’ then the fourth element is picked for processing(index value starts from 0). Similarly, if the switch value is evaluated to an element’s name then that element is picked for processing.

- Throughput Controller

The Throughput Controller is used to control the processing of its child elements in terms of the total number of executions or the percentage of execution specified in its control panel.

- Transaction Controller

The Transaction Controller is one of the widely used controller in JMeter scripts. It is used to group multiple sampler requests into one. The response time and other performance metrics of the test result are evaluated for the whole transaction. For example – while checking the performance of home page of an application, we can notice that launching the home page generates numerous requests at the backend. Hence, this requires grouping all these request into one transaction, for this, we have a transaction controller in JMeter. Once we run the script, we can find the overall response time of the whole transaction.

- While Controller

The While controller is used to run the child elements inside it till the value specified in its control panel is evaluated to false.

Listeners

Performance testing is all about analysing server responses in various forms and then presenting the same to the client.

Listeners provide pictorial representation of data gathered by JMeter about those test cases as a sampler component of JMeter is executed. It facilitates the user to view samplers result in the form of tables, graphs, trees or simple text in some log file.

Listeners can be adjusted anywhere in the test, including directly under the test plan. There are around 15 listeners provided by JMeter but mostly used ones are table, tree, and Graph.

Following is the list of all Listeners in JMeter:

Graph Results
Spline Visualizer
Assertion Results
Simple Data Writer
Monitor Results
Distribution Graph (alpha)
Aggregate Graph
Mailer Visualizer

BeanShell Listener

Summary Report

Sample Result Save Configuration

Graph Full Results

View Results Tree

Aggregate Report

View Results in Table

✧ Aggregate Graph

The Aggregate Graph listener is used to display the test results in both tabular form(reports) and graphs.

✧ Aggregate Report

The Aggregate Report listener is used to display and store test results in the form of reports.

✧ Assertion Results

The assertion results listener is used to display the assertion result for each erroneous sampler response. It is advised to not use this listener during the performance test as it is very resource-intensive. It should be used while debugging and functional testing only.

✧ Backend Listener

The backend listener is a special type of asynchronous listener used specifically with BackendListenerClient for its customization.

✧ BeanShell Listener

The BeanShell listener is used to enable BeanShell scripting in JMeter. For details on BeanShell Scripting check our tutorial Bean Shell Scripting in JMeter.

- ✧ BSF Listener

The BeanShell listener is used to enable BSF scripting in JMeter.

- ✧ Comparison Assertion Visualizer

The Comparison Assertion Visualizer is used to provide a comparison between assertion result in an easy to compare UI.

- ✧ Generate Summary Results

The Generate Summary results listener is used to store and display detailed test results to log files.

- ✧ Graph Results

The Graph results listener is used to display each sampler request's response time graph in terms of average, median, deviation, and throughput.

- ✧ JSR223 Listener

The JSR223 Listener is used to enable JSR223 scripting in JMeter.

- ✧ Mailer Visualizer

The Mailer Visualizer sampler is used to provide the functionality of sending customized mails in case of some specific error threshold.

- ✧ Monitor Results

This is a newly added listener in JMeter used to display and store server performance stats.

✧ Response Time Graph

The response time graph is used to provide the graphical representation of response time with time elapsed during the test run.

✧ Save Response to a file

The save response to a file listener is used to store the sampler response in a file. This listener is used while functional testing or debugging the test script.

✧ Simple Data Writer

The simple data writer listener is used to save the sampler response to a file after with different configurations to remove several unnecessary overheads.

✧ Summary Report

The summary report is used to store and display the test result in tabular form just like an aggregate report listener but consumes less memory(as per Apache JMeter).

✧ View Results Tree

This listener is used to provide and store test results for each and every individual sampler.

✧ View Results in Table

The view results in a table listener are used to display the sampler response header and response body.

Timers

When you perform any operation on a website or app, they naturally have pauses and delays. These can be simulated with Timers.

JMeter sends requests without applying any delay between each sampler/request. If you perform load/stress testing on your server without any delay, it will be overloaded. This not exactly what you want. You can add a timer element which will permit you to define a period to wait between each request.

Given below is the list of all the Timer elements provided by JMeter:
Synchronizing Timer
JSR223 Time
BeanShell Time
Gaussian Random Timer
Uniform Random Timer
Constant Throughput Timer
BSF Time
Poisson Random Time

✓ Constant Timer

The constant timer is one of the most widely used timers in JMeter. It pauses the execution of test for a specified constant amount of time.

✓ Uniform Random Timer

The uniform random timer is used to pause the test execution for a random time. The maximum value for random time can be specified along with the additional constant time with each wait.

✓ Constant Throughput Timer

The constant throughput timer is a special type of timer used to create pauses with variable amount of time while maintaining the overall throughput i.e. samples/minute.

✓ BeanShell Timer

The BeanShell timer is used to generate the delays using BeanShell scripting.

✓ BSF Timer

The BSF timer is used to generate delays using BSF scripting.

✓ Gaussian Random Timer

The Gaussian random timer is used to generate delays using Gaussian distribution.

✓ Synchronizing Timer

The Synchronizing timer is used to insert delays in the script by bocking a certain number of threads and when the blocked thread count reaches a specified number then the threads are released at once. Since, the synchronizing timer generates a large amount of instant load hence, it is used for spike testing.

✓ Poisson Random Timer

The Poisson random timer is used to generate the delays using Poisson distribution.

✓ JSR223 Timer

The JSR223 timer is used to generate the delays using JSR223 scripting.

Configuration Elements

Working of configuration elements is quitesimilar to those of samplers.However, it does not send requests but it allows you to modify the requests made by the samplers.

It is a simple element where you can collects the corporate configuration values of all samplers like webserver's hostname or database url etc.

A configuration element is accessible from only inside the branch where you place the element.

Given below is the list of some of the most commonly used configuration elements provided by JMeter:

Java Request Defaults
LDAP Request Defaults
LDAP Extended Request Defaults
Keystore Configuration
JDBC Connection Configuration
Login Config Element
CSV Data Set Config
FTP Request Defaults
TCP Sampler Config
User Defined Variables
HTTP Authorization Manager
HTTP Cache Manager
HTTP Cookie Manager
HTTP Proxy Server
HTTP Request Defaults

HTTP Header Manager

Simple Config Element

Random Variable

◆ CSV Data Set Config

The CSV Data Set Config is used to read data from CSV file, put the data into variable(s) and then use the variable(s) in the sampler requests. Check our tutorial on Parameterization in JMeter for details.

◆ HTTP Cache Manager

The HTTP Cache manager is used in test scripts to add the Caching functionalities of web applications. This element is just required to be added at the same level or higher than the sampler request where caching functionality is required.

◆ HTTP Cookie Manager

The HTTP Cookie manager is required for session handling by providing the functionality of storing and sending of cookies.

◆ User Defined Variables

As the name suggests, the User Defined Variable config element is used to create variables with a value (key-value pairs) that are used across the test script.

◆ Random Variable

The random variable config element is used to generate random numeric values within a range of specified minimum and maximum values.

◆ Counter

The counter config element is used to create a variable that gets incremented by specified value in each iteration within a range of minimum and maximum values.

◆ JDBC Connection Configuration

JDBC Connection Configuration is used with JDBC request sampler to create JDBC connection settings.

◆ FTP Request Defaults

The FTP request defaults are used to create default settings while testing FTP servers.

◆ DNS Cache Manager

The DNS Cache Manager is used while testing applications behind the load-balancers.

◆ HTTP Authorization Manager

The HTTP Authorization Manager is used for testing applications requiring multiple logins for ensuring authorization.

◆ HTTP Request Defaults

The HTTP Request Defaults config element is used for setting default values for HTTP requests.

◆ HTTP Header Manager

The HTTP Header Manager is used to override the HTTP request headers.

◆ Java Request Defaults

The Java request defaults config elements are used to specify default values for Java Request sampler.

◆ Keystore Configuration

The Keystore Configuration config element is used to configure the loading of keystores.

◆ Login Config Element

The logic Config element is used to create default credentials for the samplers using username and password in their setup

◆ LDAP Request Defaults

The LDAP request defaults are used to create default settings while testing LDAP servers.

◆ LDAP Extended Request Defaults

The LDAP extended request defaults are used to create default settings for LDAP Extended Request samplers.

◆ TCP Sampler Config

The TCP Sampler Config is used for creating default settings for TCP Sampler.

◆ Simple Config Element

The simple config element is used to create key-value pairs that can be used across the test script.

Pre-processor Elements

A Pre-Processor element is executed just before the request made by the sampler. If a Pre-processor is attached to a sampler element then it will execute just prior to that sampler element running.

A Pre-processor element is used to modify the settings of a sample request just before it runs, or to update variables that are not extracted from response text.

Following is a list of all Pre-processor elements provided by JMeter:

JDBC Pre-processor
JSR223 Pre-processor
RegEx User Parameters
BeanShell Pre-processor
BSF Pre-processor
HTML Link Parser
HTTP URL Re-writing Modifier
HTTP User Parameter Modifier
User Parameters

- BeanShell PreProcessor

The BeanShell preprocessor is used to perform some operation using beanshell scripting before a sampler request.

- HTML Link Parser

The HTML Link Parser is used to extract links from HTML response fetched from server.

- HTTP URL Re-writing Modifier

The HTTP URL Re-writing modifier can be added at either Thread Group level or Sampler level to fetch a specific sessionID parameter from the response of a request and then use the parameter's value on other requests.

- ❖ JSR223 PreProcessor

The JSR223 preprocessor is used to perform some operation using JSR223 scripting before a sampler request.

- ❖ JDBC PreProcessor

The JDBC PreProcessor is used to execute certain specified SQL queries before a sampler request processing.

- ❖ RegEx User Parameters

The RegEx user parameters are used to extract HTTP parameters from the certain requests using a regular expression and then passed as a request parameter to other sampler requests.

- ❖ User Parameters

The User parameters are used to specify values for User variables used within Thread Groups.

Post-processor Elements

A Post-processor element is executed after a sampler request has been made. If a Post-Processor is attached to a Sampler element then it will execute just after that sampler element runs.

A Post-processor is most often used to process the response data, for example, to extract a particular value for future purpose.

Given below is the list of all Post-processor elements provided by JMeter:

CSS/JQuery Extractor

BeanShell Post-processor

JSR223 Post-processor

JDBC Post-processor

Debug Post-processor

Regular Expression Extractor

XPath Extractor

Result Status Action Handler

BSF Post-processor

- Regular Expression Extractor

Regular Expression Extractor is the most frequently used post-processor. It is used to extract values from HTTP response using regular expression and to store the values to a variable. Then the variable can either be used as a request parameter to other samplers or can be used for other purposes like assertions, debugging, storing in the file, etc.

- BeanShell PostProcessor

The BeanShell PostProcessor is used to perform some operation using BeanShell scripting after a sampler request processing.

- CSS/JQuery Extractor

The CSS/JQuery Extractor is used to extract values from HTTP response using CSS or JQuery expressions and then the extracted value is stored in a variable.

- XPath Extractor

The XPath Extractor is used to extract values from HTTP response using CSS or JQuery expressions and then the extracted value is stored in a variable

❖ Debug PostProcessor

The Debug PostProcessor is used to create a subSample having properties of previous sampler requests, JMeter properties, JMeter variables or system properties.

❖ JSR223 PostProcessor

The JSR223 PostProcessor is used to perform some operation using JSR223 scripting after a sampler request processing.

❖ JDBC PostProcessor

The JDBC PostProcessor is used to execute certain specified SQL queries after a sampler request processing.

❖ JSON Path PostProcessor

The JDBC PostProcessor is used to extract data from JSON response using JSON-Path syntax.

❖ Result Status Action Handler

The Result Status Action Handler is used to stop a thread group or the whole test in case of a specific sampler failure.

Assertions

Assertions in JMeter are the test plan elements that are used to validate the response received from server for a particular sampler request. In order to test a sampler response, we can add different assertions to the sampler requests as child. If an assertion fails, the sampler request is marked as failed and the same gets reflected in the test results listeners like – aggregate report listener

- Response Assertion

The response assertion used in test scripts to validate a pattern in the response body, header, code, message etc. There are different pattern matching rules to validate the response like-

Contains – If the response text contains the regular expression to be matched
Matches – If the whole response text matches the regular expression
Equals – If the whole response text matches the pattern(not regular expression but the pattern string)
Substring – If the response text contains the pattern(not regular expression)
Not – To check that the pattern is not present in the response text

- HTML Assertion

The HTML assertion is used to check the HTML syntax of the response.

- Size Assertion

The size assertion is used to validate the size of the response with a specified value in bytes.

■ Compare Assertion

The Compare Assertion is used to compare sampler results.

■ BSF Assertion

The BSF Assertion is used to validate the sampler result using BSF scripting.

■ Duration Assertion

The duration assertion is used to validate that the sampler request gets processed within a specified amount of time.

■ XML Assertion

The XML assertion is used to validate that the response follows a valid XML syntax.

■ XML Schema Assertion

The XML Schema Assertion is used to validate the response against a specified XML schema.

■ XPath Assertion

The XPath assertion is used to validate the response using XPath expressions.

■ MD5Hex Assertion

The MD5Hex Assertion is used to validate the sampler result by checking its MD5Hex hashcode against a hashcode value provided.

(MD5 (Message Digest Method 5) is a cryptographic hash algorithm used to generate a 128-bit digest from a string of any length. It represents the digests as 32 digit hexadecimal numbers.)

- SMIME Assertion

The SMIME Assertion is used to validate the body of a MIME message.

Secure Multipurpose Internet Mail Extensions (S/MIME)

- JSR223 Assertion

The JSR223 Assertion is used to validate the sampler result using JSR223 scripting.