**Test Plan:**

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

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

The Test Plan object has a checkbox called "Functional Testing". If selected, it will cause JMeter to record the data returned from the server for each sample. If you have selected a file in your test listeners, this data will be written to file. This can be useful if you are doing a small run to ensure that JMeter is configured correctly, and that your server is returning the expected results. The consequence is that the file will grow huge quickly, and J Meter's performance will suffer. This option should be off if you are doing stress-testing (it is off by default).

If you are not recording the data to file, this option makes no difference.

You can also use the Configuration button on a listener to decide what fields to save.

*Advantage of using Web test plan in the script:*

Another positive feature available to build test plans is to make whole process a breeze. To test a SOAP interface layer all that is required is the URL and SOAP request. It is possible to develop the test plan starting with these basics and add on variables, counters, parameters, CSV files, loops, logs, and anything else. There are virtually no limits in designing the test and making it as maintainable as possible.

Managing and maintaining JMeter scripts can be challenging unless the user knows to use the tool effectively. But here too, help is on hand. The JMeter website is lush with documentation of all hues, including wikis, user manual, docs and user experiences. The user manual describes how to create specific test plans for web, database, JMS, Web service and other options, complete with step-by-step instructions, bitmaps, examples, best practices, tips, samples of regular expressions, step-by-step guides on how to use the variables and predefined functions, and component reference that describes in detail how every component can be used.

How to add a thread group to the script:

Thread group elements are the beginning points of any test plan. All controllers and samplers must be under a thread group. Other elements, e.g. Listeners, may be placed directly under the test plan, in which case they will apply to all the thread groups. As the name implies, the thread group element controls the number of threads JMeter will use to execute your test. The controls for 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

Each thread will execute the test plan in its entirety and completely independently of other test threads. Multiple threads are used to simulate concurrent connections to your server application.

The ramp-up period tells JMeter how long to take to "ramp-up" to the full number of threads chosen. If 10 threads are used, and the ramp-up period is 100 seconds, then JMeter will take 100 seconds to get all 10 threads up and running. Each thread will start 10 (100/10) seconds after the previous thread was begun. If there are 30 threads and a ramp-up period of 120 seconds, then each successive thread will be delayed by 4 seconds.

Ramp-up needs to be long enough to avoid too large a work-load at the start of a test, and short enough that the last threads start running before the first ones finish unless one wants that to happen.

Start with Ramp-up = number of threads and adjust up or down as needed.

By default, the thread group is configured to loop once through its elements.

Version 1.9 introduces a test run scheduler. Click the checkbox at the bottom of the Thread Group panel to reveal extra fields in which you can enter the start and end times of the run. When the test is started, JMeter will wait if necessary until the start-time has been reached. At the end of each cycle, JMeter checks if the end-time has been reached, and if so, the run is stopped, otherwise the test is allowed to continue until the iteration limit is reached.

Alternatively, one can use the relative delay and duration fields. Note that delay overrides start-time, and duration over-rides end-time.