

# UserManual BuildWebTest

**h1. Building a Web Test Plan** In this section, you will learn how to create a basic Test Plan to test a Web site. You will create five users that send requests to two pages on the Jakarta Web site. Also, you will tell the users to run their tests twice. So, the total number of requests is (5 users) x (2 requests) x (repeat 2 times) = 20 HTTP requests. To construct the Test Plan, you will use the following elements: Thread Group , HTTP Request , HTTP Request Defaults , and Graph Results . For a more advanced Test Plan, see [Building an Advanced Web Test Plan]../BuildAdvancedTest]. {anchor:users} h2. Adding Users The first step you want to do with every JMeter Test Plan is to add a Thread Group element. The Thread Group tells JMeter the number of users you want to simulate, how often the users should send requests, and the how many requests they should send. Go ahead and add the [ThreadGroup] element by first selecting the Test Plan, clicking your right mouse button to get the Add menu, and then select Add -> [ThreadGroup]. You should now see the Thread Group element under Test Plan. If you do not see the element, then "expand" the Test Plan tree by clicking on the Test Plan element. Next, you need to modify the default properties. Select the Thread Group element in the tree, if you have not already selected it. You should now see the Thread Group Control Panel in the right section of the JMeter window (see Figure 5.1 below) {{http://jakarta.apache.org/jmeter/images/screenshots/webtest/threadgroup.png}} ^Figure 5.1. Thread Group with Default Values^ Start by providing a more descriptive name for our Thread Group. In the name field, enter Jakarta Users. Next, increase the number of users (called threads) to 5. In the next field, the Ramp-Up Period, leave the the default value of 0 seconds. This property tells JMeter how long to delay between starting each user. For example, if you enter a Ramp-Up Period of 5 seconds, JMeter will finish starting all of your users by the end of the 5 seconds. So, if we have 5 users and a 5 second Ramp-Up Period, then the delay between starting users would be 1 second (5 users / 5 seconds = 1 user per second). If you set the value to 0, then JMeter will immediately start all of your users. Finally, clear the checkbox labeled "Forever", and enter a value of 2 in the Loop Count field. This property tells JMeter how many times to repeat your test. If you enter a loop count value of 0, then JMeter will run your test only once. To have JMeter repeatedly run your Test Plan, select the Forever checkbox. {{In most applications, you have to manually accept changes you make in a Control Panel. However, in JMeter, the Control Panel automatically accepts your changes as you make them. If you change the name of an element, the tree will be updated with the new text after you leave the Control Panel (for example, when selecting another tree element).}} {{http://jakarta.apache.org/jmeter/images/screenshots/webtest/threadgroup2.png}} ^Figure 5.2. Jakarta Users Thread Group^ {anchor:defaults} h2. Adding Default HTTP Request Properties Now that we have defined our users, it is time define the tasks that they will be performing. In this section, you will specify the default settings for your HTTP requests. And then, in section 5.3, you will add HTTP Request elements which use some of the default settings you specified here. Begin by selecting the Jakarta Users element. Click your right mouse button to get the Add menu, and then select Add -> Config Element -> HTTP Request Defaults. Then, select this new element to view its Control Panel (see Figure 5.3). {{http://jakarta.apache.org/jmeter/images/screenshots/webtest/http-defaults1.png}} ^Figure 5.3. HTTP Request Defaults^ Like most JMeter elements, the HTTP Request Defaults Control Panel has a name field that you can modify. In this example, leave this field with the default value. Skip to the next field, which is the Web Server's Server Name/IP. For the Test Plan that you are building, all HTTP requests will be sent to the same Web server, jakarta.apache.org. Enter this domain name into the field. This is the only field that we will specify a default, so leave the remaining fields with their default values. {{The HTTP Request Defaults element does not tell JMeter to send an HTTP request. It simply defines the default values that the HTTP Request elements use.}} {{http://jakarta.apache.org/jmeter/images/screenshots/webtest/http-defaults2.png}} ^Figure 5.4. HTTP Defaults for our Test Plan^ {anchor:cookies} h2. Adding Cookie Support Nearly all web testing should use cookie support, unless your application specifically doesn't use cookies. To add cookie support, simply add an HTTP Cookie Manager to each Thread Group in your test plan. This will ensure that each thread gets its own cookies, but shared across all HTTP Request objects. To add the HTTP Cookie Manager , simply select the Thread Group , and choose Add -> Config Element -> HTTP Cookie Manager, either from the Edit Menu, or from the right-click pop-up menu. {anchor:requests} h2. Adding HTTP Requests In our Test Plan, we need to make two HTTP requests. The first one is for the Jakarta home page (http://jakarta.apache.org/), and the second one is for the Project Guidelines page (http://jakarta.apache.org/site/guidelines.html). {{JMeter sends requests in the order that they appear in the tree.}} Start by adding the first HTTP Request to the Jakarta Users element (Add -> Sampler -> HTTP Request). Then, select the HTTP Request element in the tree and edit the following properties (see Figure 5.5): # Change the Name field to "Home Page". # Set the Path field to "/". Remember that you do not have to set the Server Name field because you already specified this value in the HTTP Request Defaults element. \ {{http://jakarta.apache.org/jmeter/images/screenshots/webtest/http-request1.png}} ^Figure 5.5. HTTP Request for Jakarta Home Page^ \ Next, add the second HTTP Request and edit the following properties (see Figure 5.6): # Change the Name field to "Project Guidelines". # Set the Path field to "/site/guidelines.html". \ {{http://jakarta.apache.org/jmeter/images/screenshots/webtest/http-request2.png}} ^Figure 5.6. HTTP Request for Jakarta Project Guidelines Page^ {anchor:listener} h2. Adding a Listener to View Store the Test Results The final element you need to add to your Test Plan is a Listener . This element is responsible for storing all of the results of your HTTP requests in a file and presenting a visual model of the data. Select the Jakarta Users element and add a Graph Results listener (Add -> Listener -> Graph Results). Next, you need to specify a directory and filename of the output file. You can either type it into the filename field, or select the Browse button and browse to a directory and then enter a filename. {{http://jakarta.apache.org/jmeter/images/screenshots/graph\_results.png}} ^Figure 5.7. Graph Results Listener^ {anchor:saving} h2. Saving the Test Plan Although it is not required, we recommend that you save the Test Plan to a file before running it. To save the Test Plan, select Save Test Plan from the File menu (with the latest release, it is no longer necessary to select the Test Plan element first). JMeter allows you to save the entire Test Plan tree or only a portion of it. To save only the elements located in a particular "branch" of the Test Plan tree, select the Test Plan element in the tree from which to start the "branch", and then click your right mouse button to access the Save As menu item. Alternatively, select the appropriate Test Plan element and then select Save As from the Edit menu. {anchor:running} h2. Running the Test Plan From the Run menu, select Run. JMeter lights up a green square in the upper-right-hand corner to indicate if a test is currently running. The square is turned gray when all tests stop. Even after you select "stop", the green light will stay on until all test threads have exited. Once JMeter has finished running your Test Plan, select Stop from the Run menu. If you selected a file to save the results to in your listener, then you will have a file that can be opened in any visualizer. Each visualizer will display the results in it's own fashion. It is possible to have the same file open in more than one visualizer. This is not a problem. JMeter will ensure during the test run that no sample is recorded to the same file more than once.