

UserManual BuildDbTest

Building a Database Test Plan

In this section, you will learn how to create a basic Test Plan to test a database server. You will create ten users that send two SQL requests to the database server. Also, you will tell the users to run their tests three times. So, the total number of requests is (10 users) x (2 requests) x (repeat 3 times) = 60 JDBC requests. To construct the Test Plan, you will use the following elements: Thread Group , JDBC Request , Graph Results .

This example uses the PostgreSQL `org.postgresql.Driver` database driver. To use this driver, its containing `.jar` file must be copied to the extension `.../lib/` directory (see JMeter's Classpath for more details). Otherwise, expect a substantial amount of stack traces when running this test plan.

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 7.1 below)

<http://jakarta.apache.org/jmeter/images/screenshots/webtest/threadgroup.png>

Figure 7.1. Thread Group with Default Values

Start by providing a more descriptive name for our Thread Group. In the name field, enter JDBC Users.

You will need a valid database, database table, and user-level access to that table. In the example shown here, the database is 'mydb' and the tables' name is 'Stocks'.

Next, increase the number of users (called threads) to 10.

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 3 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).

See Figure 7.2 for the completed JDBC Users Thread Group.

<http://jakarta.apache.org/jmeter/images/screenshots/jdbctest/threadgroup2.png>

Figure 7.2. JDBC Users Thread Group

Adding JDBC Requests

Now that we have defined our users, it is time to define the tasks that they will be performing. In this section, you will specify the JDBC requests to perform.

Although there are JDBC Database Connection Pool Defaults , JDBC Database Login Defaults , and JDBC SQL Query Defaults components, they will not be used in this example.

Begin by selecting the JDBC Users element. Click your right mouse button to get the Add menu, and then select Add --> Sampler --> JDBC Request. Then, select this new element to view its Control Panel (see Figure 7.3).

<http://jakarta.apache.org/jmeter/images/screenshots/jdbctest/JDBCRequest.png>

Figure 7.3. JDBC Request

In our Test Plan, we will make two JDBC requests. The first one is for Eastman Kodak stock, and the second is Pfizer stock (obviously you should change these to examples appropriate for your particular database). These are illustrated below.

JMeter sends requests in the order that you add them to the tree.

Start by editing the following properties (see Figure 7.4):

1. Change the Name to "Kodak".
2. Enter the JDBC URL field.
3. Enter the Driver Class field.
4. Change the Number of Connections in Pool field to "1".
5. Change the Max. Usage For Each Connection field to "1".
6. Enter the Username field.
7. Enter the Password field.
8. Enter the SQL Query String field.

<http://jakarta.apache.org/jmeter/images/screenshots/jdbctest/JDBCRequest2.png>

Figure 7.4. JDBC Request for Eastman Kodak stock

Next, add the second JDBC Request and edit the following properties (see Figure 7.5):

1. Change the Name to "Pfizer".
2. Enter the JDBC URL field.
3. Enter the Driver Class field.
4. Change the Number of Connections in Pool field to "1".
5. Change the Max. Usage For Each Connection field to "1".
6. Enter the Username field.
7. Enter the Password field.
8. Enter the SQL Query String field.

<http://jakarta.apache.org/jmeter/images/screenshots/jdbctest/JDBCRequest3.png>

Figure 7.6. JDBC Request for Pfizer stock

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 JDBC requests in a file and presenting a visual model of the data.

Select the JDBC Users element and add a Graph Results listener (Add --> Listener --> Graph Results).

<http://jakarta.apache.org/jmeter/images/screenshots/jdbctest/graph-results.png>

Figure 7.6. Graph results Listener

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.

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.