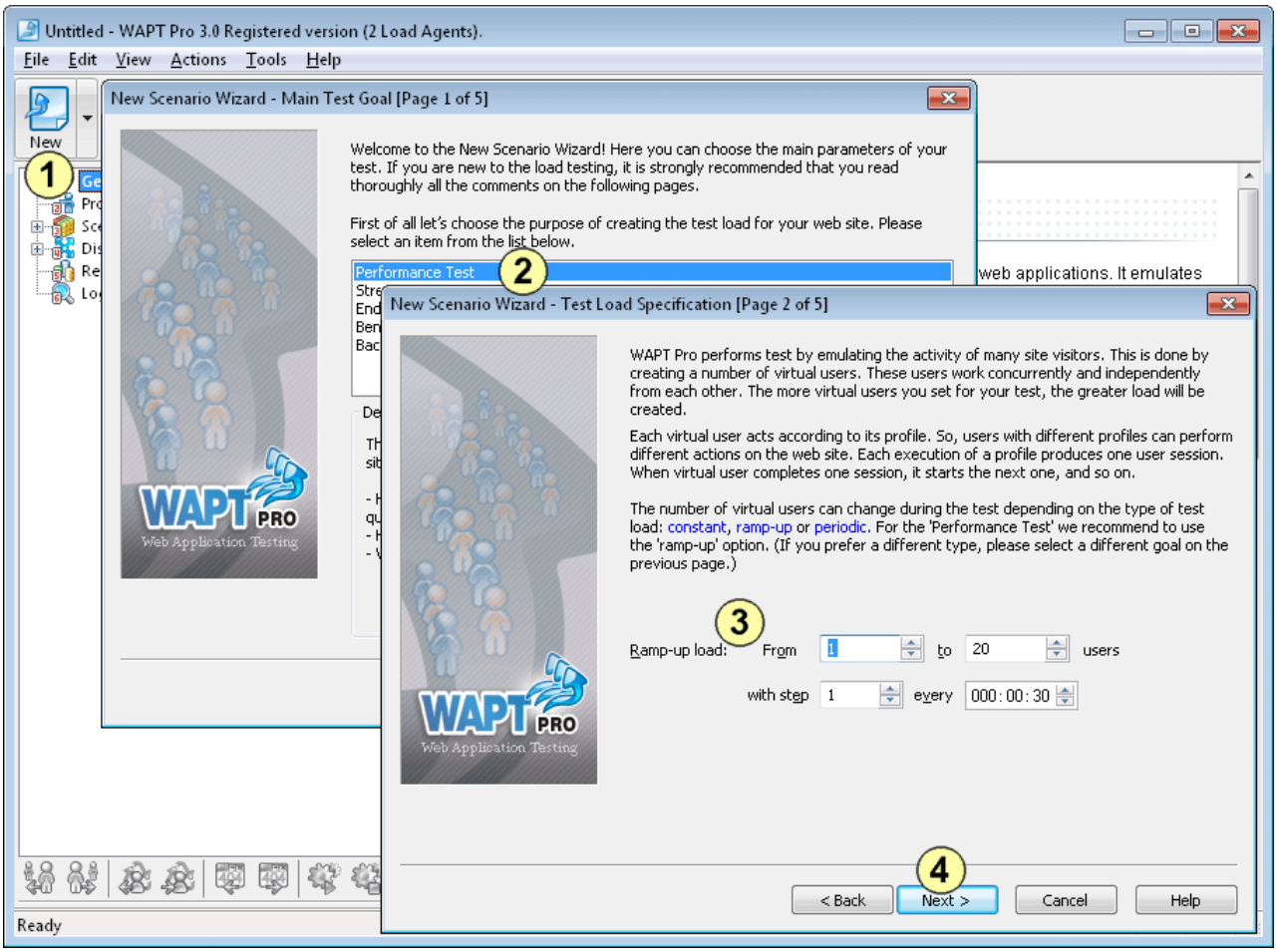
# Creating test scenario

Your work with WAPT starts with creating a test scenario. Here you can choose your test objectives and specify basic parameters of your test, such as the number of virtual users, type of load, test duration, etc.

1 Click the **"New"** button on the toolbar to create a new scenario. This will launch the **New Scenario Wizard**.



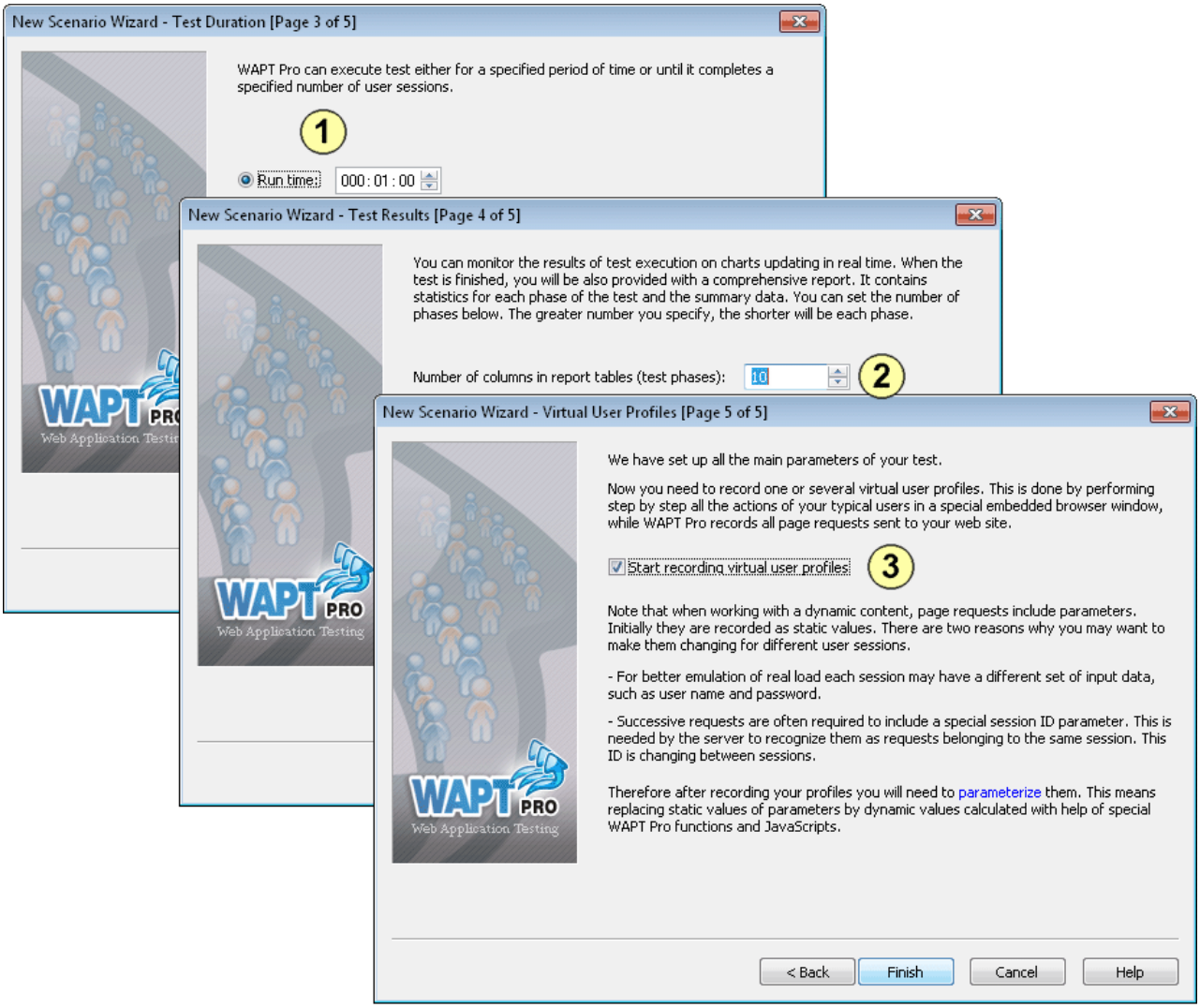

2 In this example we will create a simple performance test, so choose the "Performance Test" option on the first page of the Wizard. Click **"Next"** to continue.

3 On Page 2 you can see the recommended type of load for your test and specify its parameters. For performance testing it is recommended to use ramp-up load. This means that the number of virtual users will grow during the test, so that you can compare the web site performance on different test phases depending on the changing load.

4 Click the **"Next"** button to proceed to Page 3.

# Test duration and report options

1 On Page 3 you can choose test duration. You can either specify exact time for the test or set the total number of virtual user sessions that should be executed before the test completion. Now let's proceed to Page 4.




2 Here you can specify the number of test phases for which you want to obtain performance data. Each phase will be represented by a separate column in the report that will be created when the test is completed. The more columns you specify, the more detailed information you will see in the report. Note, however that if you specify a big number, the report will take more time to generate, so it is recommended to leave the default setting. In any case you will be able to get the most detailed information on graphs.

3 The last page of the Wizard contains some important hints on how you should proceed with the design of your test.
Note that any options you choose in the Scenario Wizard can be changed later. To do this click the **"Test Volume"** page in the left view of the WAPT window.

The most important part of the work is the creation of profiles that will define the behavior of the virtual users. During the test each virtual user works according to its profile. One execution of the profile creates one user session. As soon the user finishes a session, it starts a new one, and so on until the test is completed. So, each virtual user emulates multiple real users visiting the site one after another.

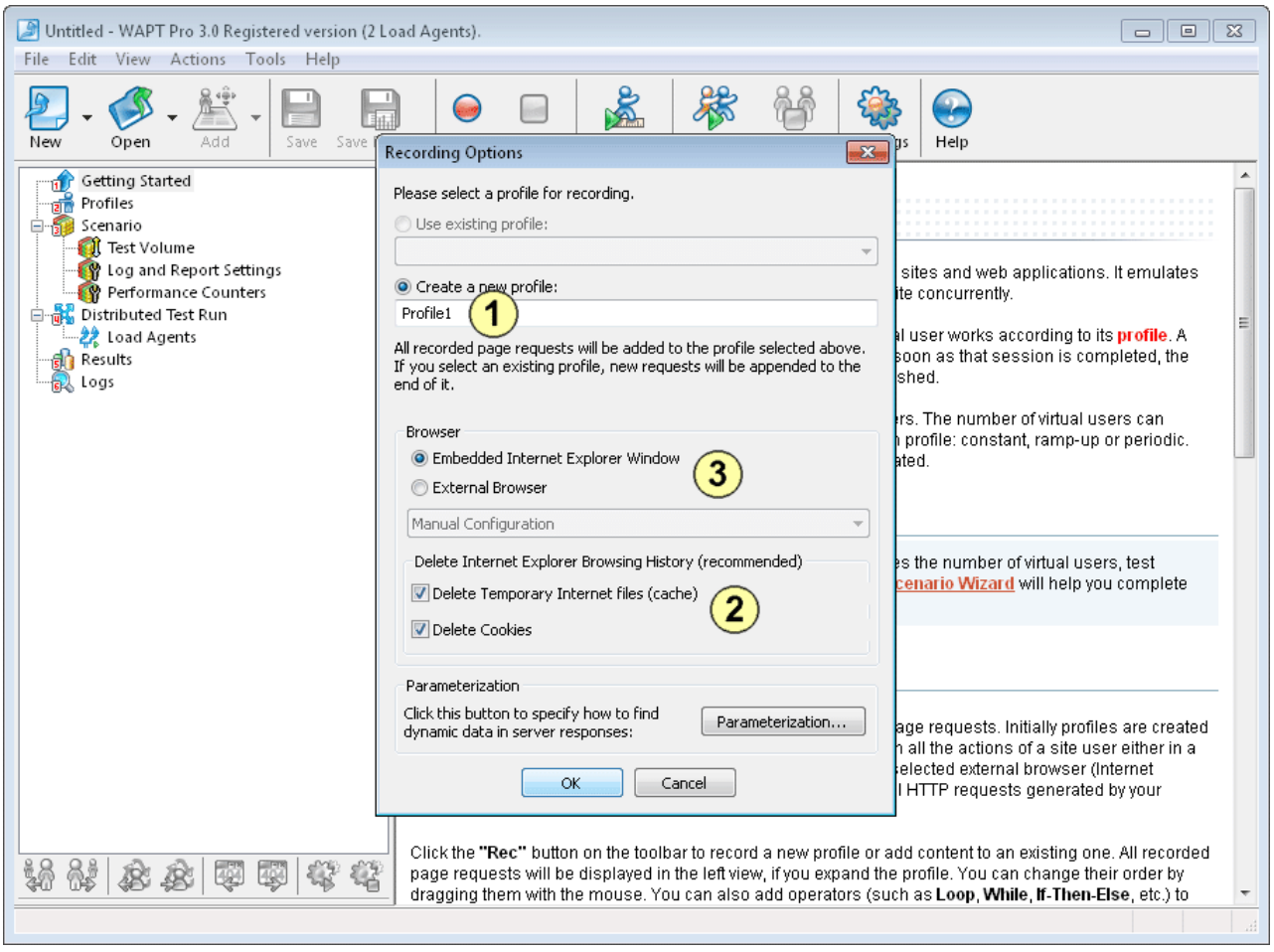
# Creating a virtual user profile

Initially profiles are recorded with help of a browser. You should simply perform step by step all the actions of the user that you want to emulate. WAPT will record everything in form of HTTP requests. During the test WAPT will execute a copy of the recorded session by sending the very similar sequence of requests. Actually, the requests will be the same with only few modifications that we will mention later.

If you want to emulate different types of users in the same test, you can record several different profiles.

Click the **"Rec"** button on the toolbar to open the **"Recording Options"** dialog. (When the Scenario Wizard is finished, that dialog is displayed automatically.)

(1) We will create a new profile now, so you need to choose a name for it. If you already had a previously recorded profile, you could select it and append the recorded requests to its end.




(2) It is recommended that you delete your Internet Explorer temporary files and cookies before recording. This is required to simulate a session from a "clean" system of a user who has never visited the site before. WAPT will do this automatically if you leave the corresponding options checked.

(3) You can either use the embedded Internet Explorer window for recording, or choose an external browser for this purpose. Embedded window is preferable in most cases, because it provides more visibility and lets WAPT better organize the recorded requests. However if you experience any problems using it (like JavaScript error messages appearing during recording), you can use an external browser instead.

# Recording a user session

1 Type the initial URL to the address bar and click the **"Go"** button (or press Enter). As you navigate through the web site inside the browser window, WAPT will record all your steps as a sequence of HTTP requests.

2 These requests will start appearing in the left view. Note that sometimes when you click a link WAPT adds several requests while the page loads. This is a result of the execution of the JavaScript code contained on the page. It can initiate sending additional requests. They can even appear when you work with the web interface controls.

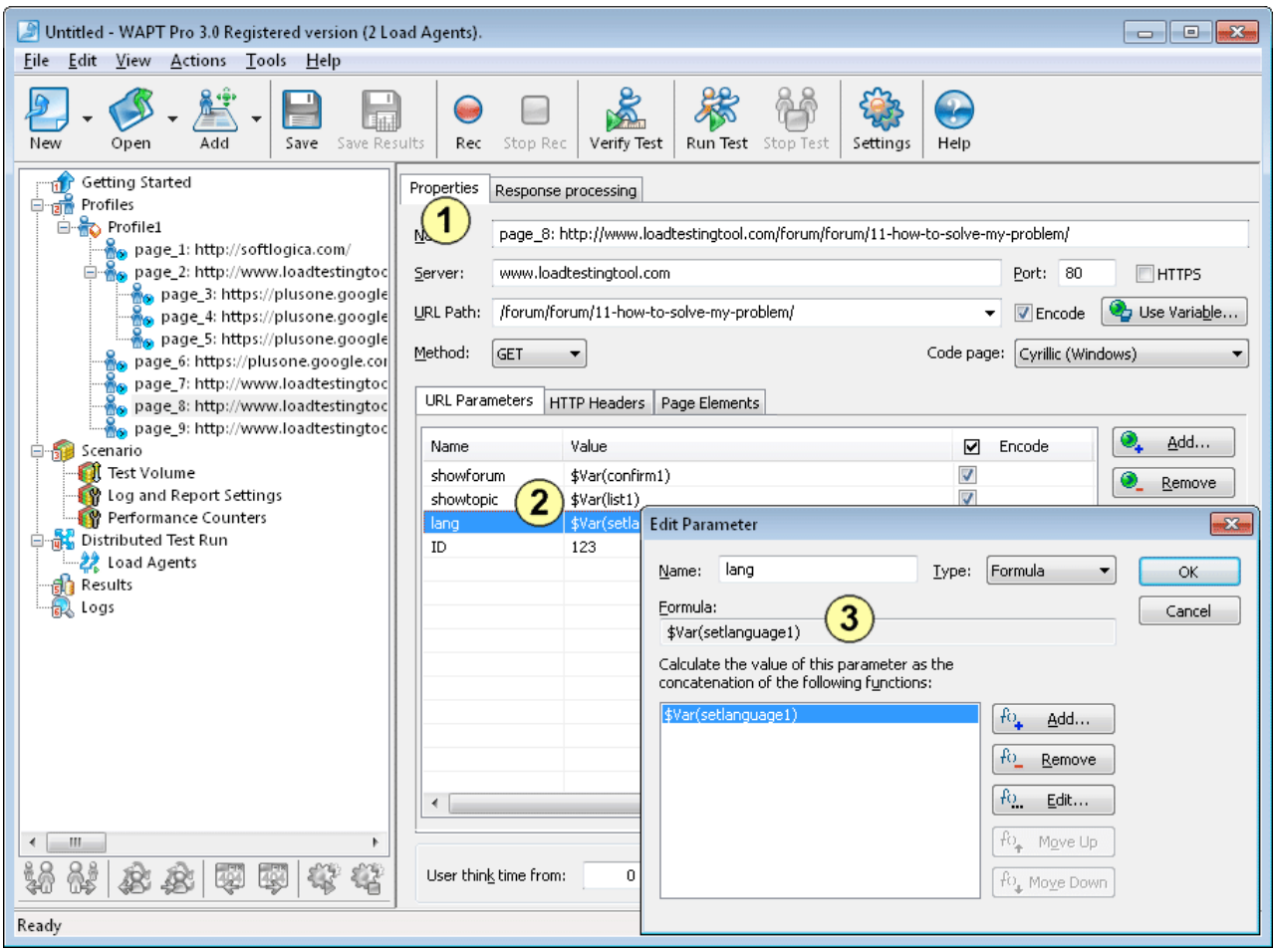
If you record using the embedded browser, WAPT will recognize such additional requests and will place them as sub-requests making a tree. In case of external browser all requests will be also recorded, but on the same level.


Note that additional requests to images, .js and .css files do not appear in the list. They are recorded as "page elements". You can find them in the properties of the corresponding page request.

3 When you finish recording click the **"Stop Rec"** button on the toolbar to exit the recording mode. Now you can select any request in the left view and edit its properties.

# Properties of a request and its parameterization

(1) Select a request in the left view. The right view will contain two tabs. The **"Properties"** tab is where you can view and edit the properties of the selected request.




(2) The most important property is the list of parameters. These parameters deliver session-specific data from the client part of the web application to the server. For example, if you use some user name and password to login to the web site, these credentials will be contained in the parameters of the login request.

Unfortunately there are no standard names and meanings for the parameters. Each web application can use its own custom ones to pass its specific data. So in some cases you have to research and guess which parameter contains the value that you need to modify.

Initially all parameters have static values saved during recording. However you can change this. For example, if you need each virtual user in your test to use a different name and password, you can replace corresponding static values with dynamic ones calculated with help of special WAPT **functions**. Such functions can generate random values, read values from a predefined list and take them from **variables** assigned earlier.

This is called ***"Parameterization"***. The understanding of this concept is very important for successful use of any load testing tool, including WAPT.

To edit any parameter, select it in the list and click the **"Edit"** button (or just double-click the parameter line).

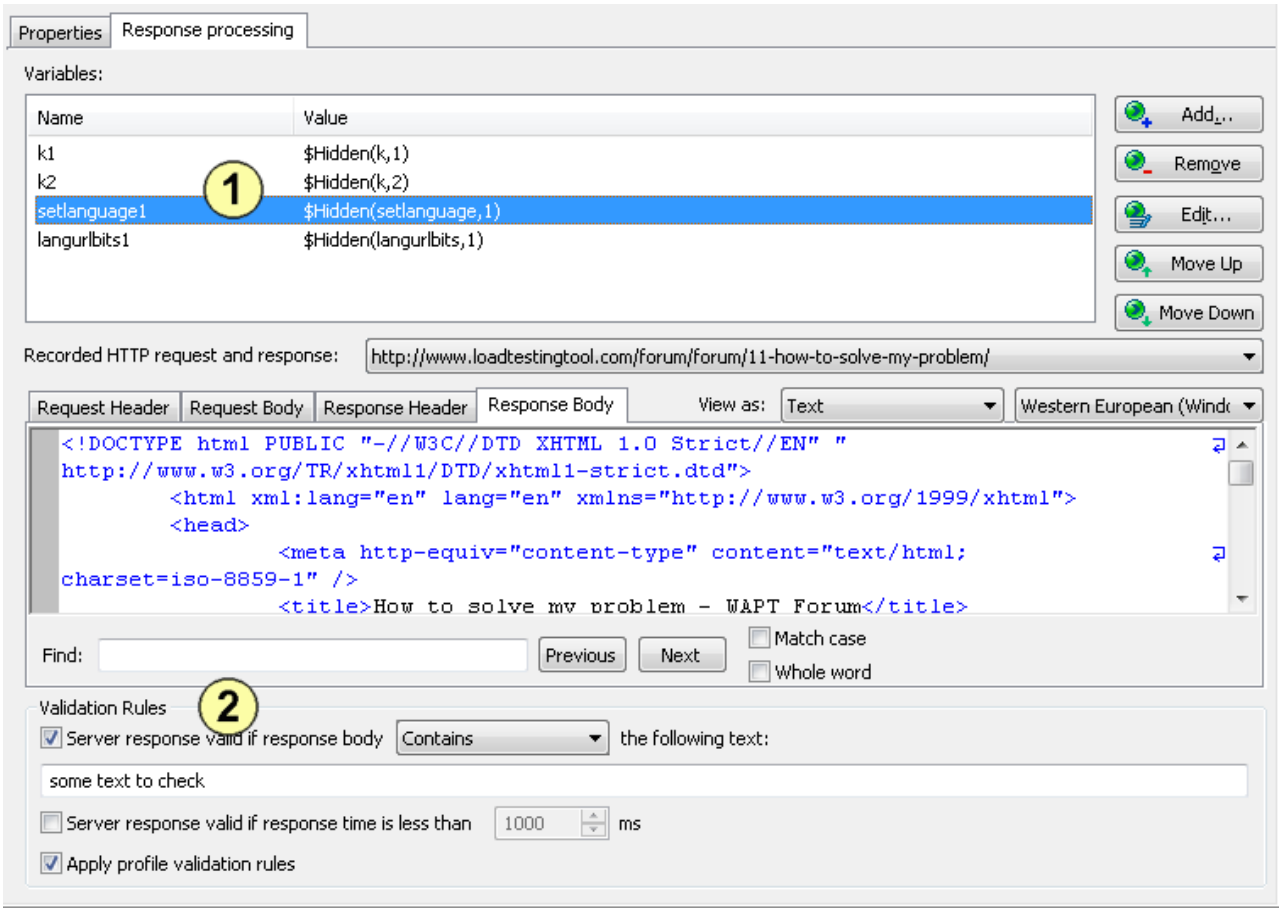
(3) In the **"Edit parameter"** dialog box you can specify how to calculate the value for the selected parameter.

# The processing of server responses

(1) Another tab available for any selected request is the **"Response processing"**. Here you can specify what to do with the received response (web page) in each emulated session. The page can contain important session-specific values that you need to use in subsequent requests.

For example, let's suppose that some page (**p1**) on your web site contains a list of items, which is different in all sessions. You select an item in the list and proceed to the next page (**p2**). At that moment your browser submits the ID of the selected item to the server as one of the parameters of the corresponding HTTP request. In fact, **p2** will be the page returned by the server in response to that request.

To emulate such session properly you need to specify how to extract that ID from the content of **p1** and insert it to the proper parameter of the request returning **p2**. This is another very important type of parameterization (it is often referred to as ***"correlation"*** in other sources).



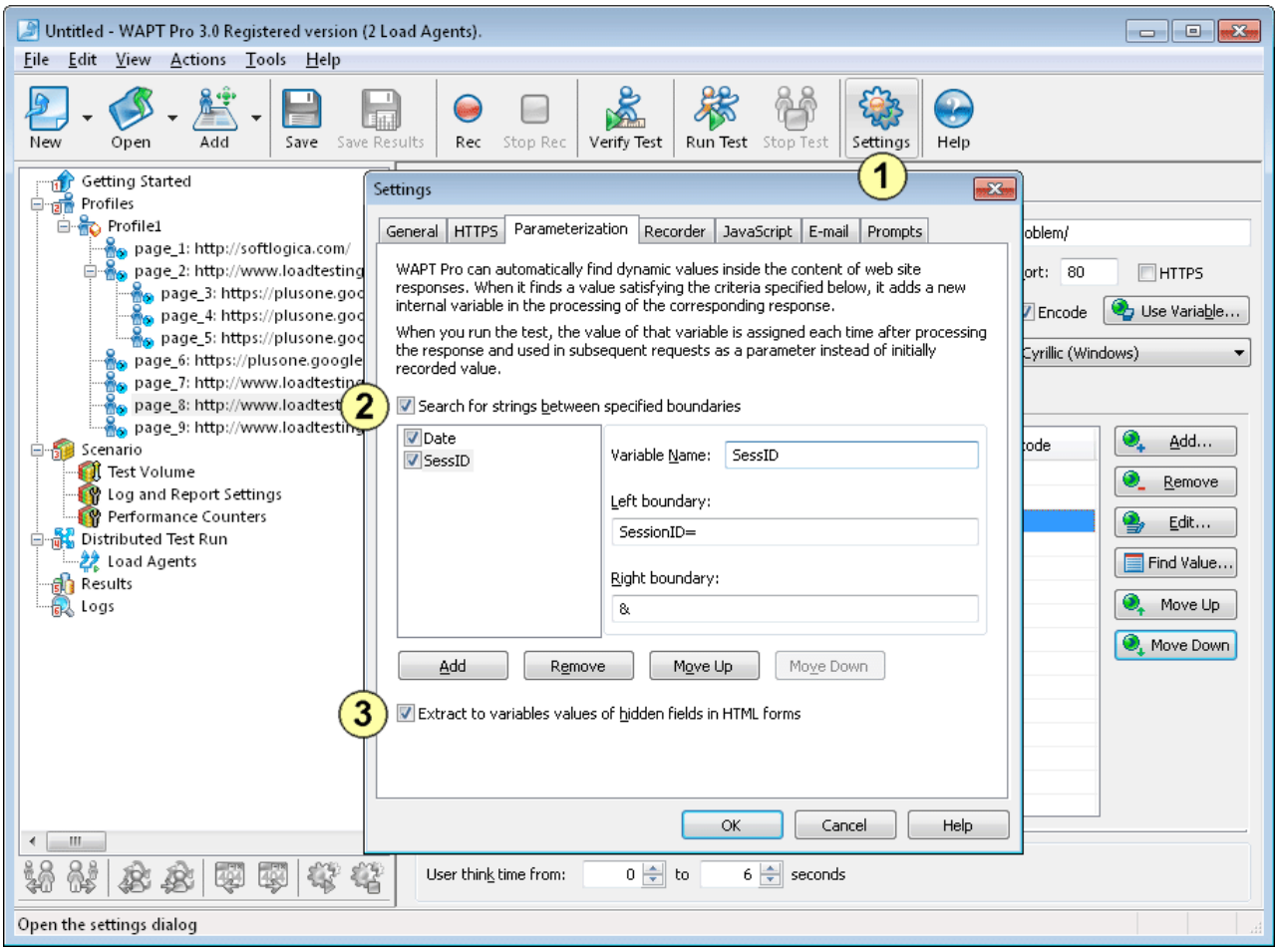

At the top of the **"Response processing"** tab there is a list of variables that will be assigned during the processing. They will store string values within user session and can be reassigned in subsequent requests. This process is very similar to the calculation of request parameters. You can use same set of functions. The difference is that variables are updated after receiving server response, whereas parameters are calculated before sending the request. Note that for your convenience the tab contains full information on the initially recorded request and response to it.

(2) There is one more goal in the processing of server responses. You can validate a response by specifying criteria that must be met in order to treat it valid. Otherwise WAPT will produce an error that will be shown in the report and logs. You can validate by response time and page content.

# Automatic parameterization

The parameterization procedure described above can be a rather complex and time consuming task. Fortunately, WAPT can do a part of this work automatically.

1 Click the **"Settings"** button on the toolbar and switch to the **"Parameterization"** tab.



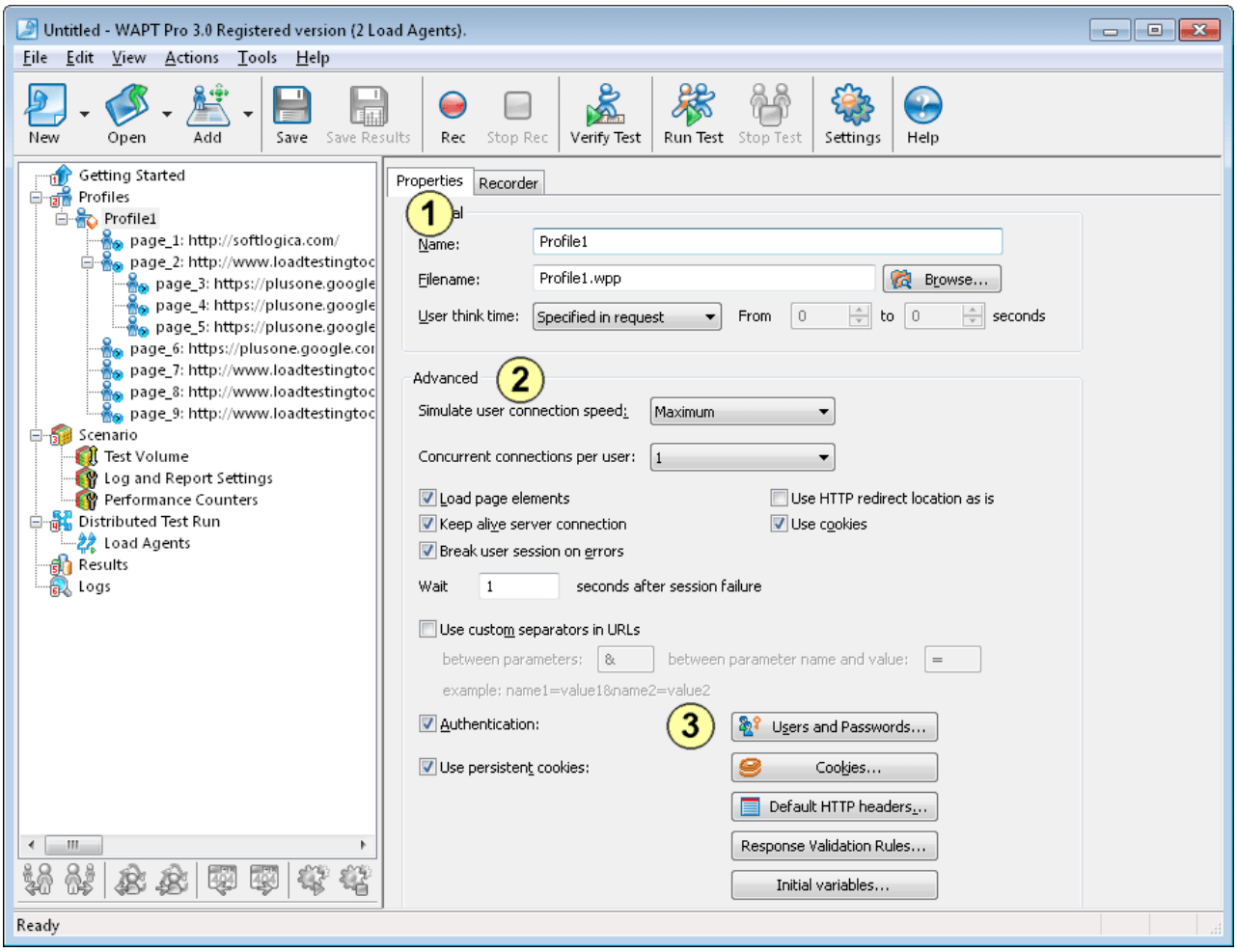

2 WAPT can check all web site responses received during recording for dynamic values located between specified left and right boundaries. When such value is found, WAPT creates a variable for it. During the test the variable will be assigned a value extracted from the actual response using same rules. WAPT will also replace that value in the parameters of all subsequent requests, so that it would be taken from the variable.

3 The "Extract to variables values of hidden fields in HTML forms" option works the same way. WAPT will create a variable for each hidden field of any HTML form found inside the server responses.

In fact, this is the most common way to pass client-server data. That is why after recording a profile you will probably see many variables created by WAPT and assigned with help of the **“$Hidden()”** function (on the “Response Processing” tab for some requests). You can also see how these variables are used in the parameters of the subsequent requests with help of the **“$Var()”** function, which returns the value of a variable.

# Properties of a virtual user profile

(1) After recording a user session you can edit other options of your profile. Select it in the left view and click the **"Properties"** tab in the right view.




(2) Here you can modify a number of options, such as “User think time” (the time between sending subsequent requests), "Load page elements", “Keep alive server connection”, etc.

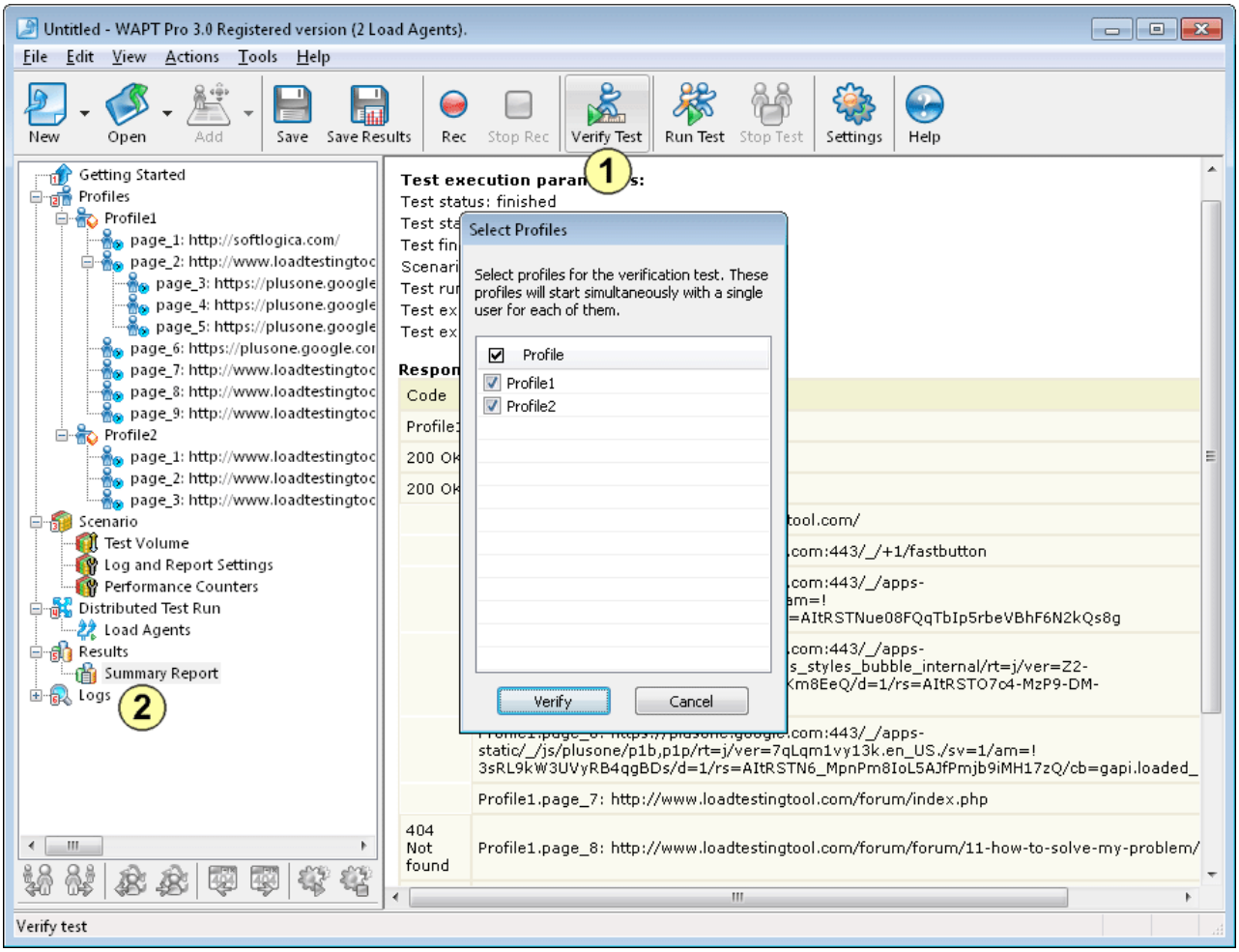
(3) If your web site requires Basic or Integrated Windows Authentication (NTLM), you can provide a list of username/password pairs that will be used one by one to login virtual users of this profile. If you provide only one username and one password, all users will share them.

Note that some options (“User think time”, “HTTP headers” and “Response validation rules”) can be reassigned in the properties of each request. For such options the values that you specify on the profile level will be the default values.

# Test verification

Since your test may require parameterization before you can execute it correctly, it is recommended that you verify it before running. This will let you check and eliminate any parameterization problems without creating a big load volume.

1 Click the **"Verify Test"** button on the toolbar. WAPT will let you select profiles for verification and will execute each of the selected profiles one time. Wait until the verification is finished. WAPT will show a simple report that contains summary information about errors occurred during the execution.




Actually what you get is a reduced version of the regular test report. It does not include data related to the load and performance, because the purpose of verification is only to check the correct work of the profiles.

Note that response codes starting with “3” (like 304) are not errors. These are HTTP redirects that are processed by WAPT automatically. Same refers to the 401 code, which means that the server requires authentication. If it is followed by the 403 code, you should provide user name and password in the profile properties and verify the test again.

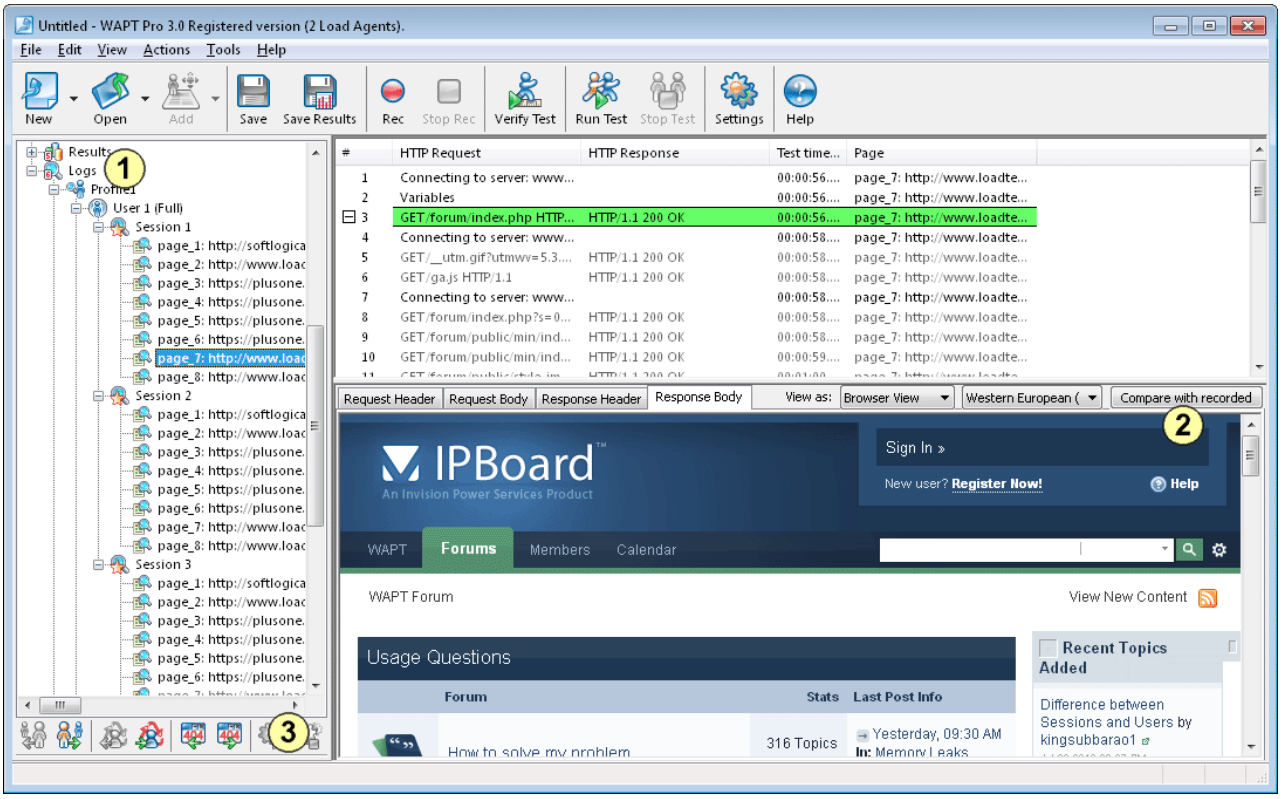
If you see the 404 (not found) code, you should check if the same code was returned when you originally recorded the profile. You can do this on the “Response processing” tab for the corresponding request. Finally, if you see the “network error”, this probably means that WAPT cannot connect to the target web site. You should check that your network configuration permits direct connection to it. You can also configure WAPT to run test through a proxy, but this is not recommended.

(2) After you verify the test, the **"Logs"** folder in the left view will become expandable. It will contain the detailed information about the executed verification sessions. To further track down all the problems you can browse the content of that folder.

# Log viewer

Log Viewer provides the detailed representation of all requests, server responses and errors appeared during the test run or verification. This information is structured with help of a tree view including profiles, virtual users, sessions and requests.

(1) Expand the **"Logs"** folder in the left view and select a page request. In the upper right view you will see the log lines containing service messages, page requests and requests to page elements. A page request is highlighted in green if it was processed without errors. Requests completed with errors are highlighted in red. You can select any line of log and see the details in the lower part of the view.




(2) When browsing logs you can compare requests and responses saved during the test run with initially recorded ones. This feature is very useful when you work on the parameterization and need to find values specific for each user session.

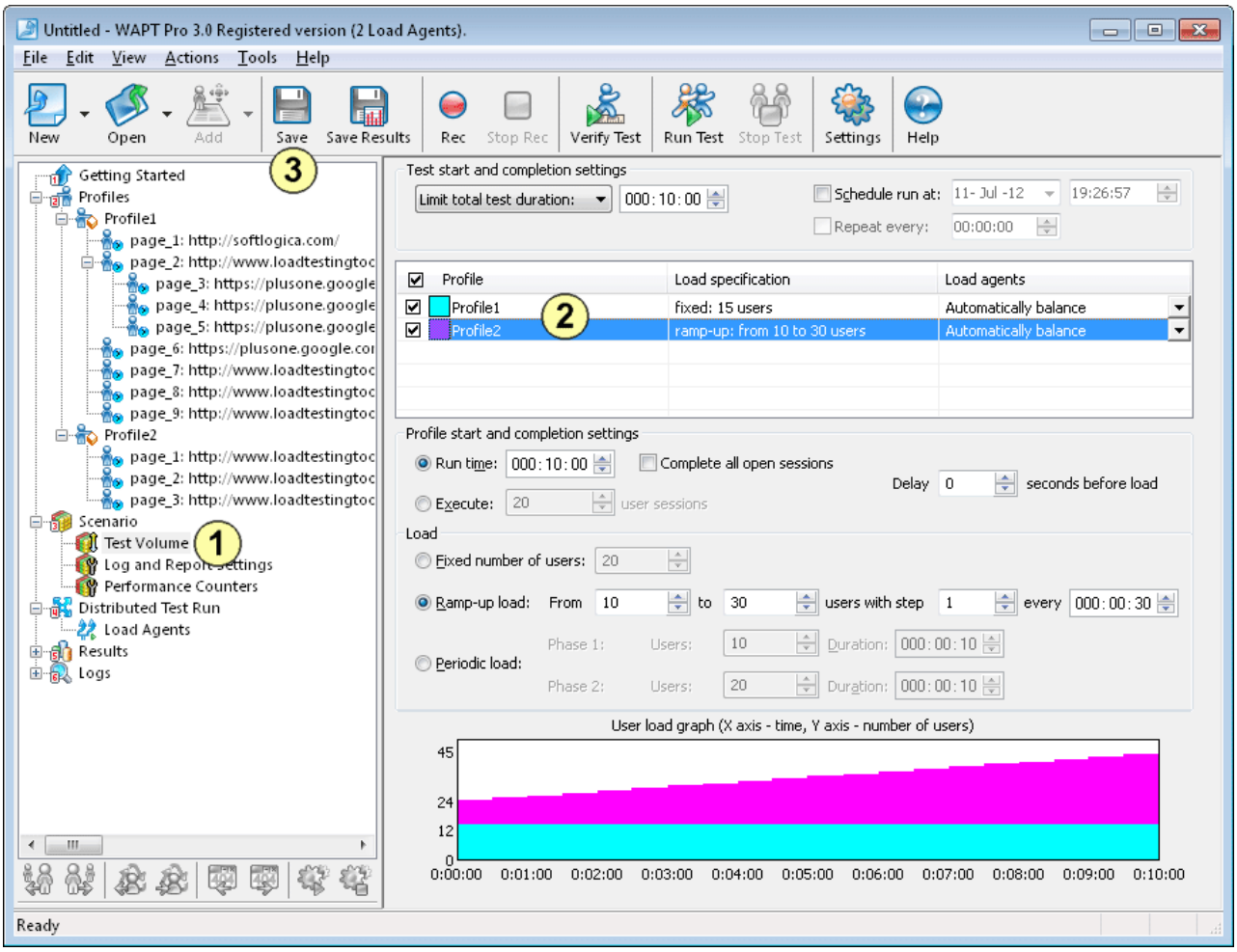
(3) You can use an additional toolbar located under the left view for fast and easy navigation through the log.

Note that by default logs are disabled for efficiency reasons, so if you want to obtain them for a test run, you should enable this feature on the **"Log and Report Settings"** page.

# Test Volume

1 Now we are ready to specify the load parameters for the test. Select the **"Test Volume"** item in the left view inside the **"Scenario"** folder.

We have already specified the type of load and test duration options in the Scenario Wizard. If you need to change them, you can do this using this page. For example, you can choose fixed number of users or periodic load instead of ramp-up.




2 Check the profiles that will be used in the test. If you have several ones, you can choose how to divide the load between them.

For each profile you can specify the type of load and other parameters. For example, you can specify fixed load of 15 virtual users for the first profile and growing load of 10 to 30 users for the second one. This is the load specified on the screenshot above. This means that the test will begin with 25 total virtual users, and during the test the load will grow to 45 users. The graph at the bottom of the page represents this situation.

You can select any profile in the list to change its load options. On the above screenshot the “Profile2” is selected, so its options are shown in the controls below the list.

In the Pro version of the product you can also specify different load agents for different profiles.

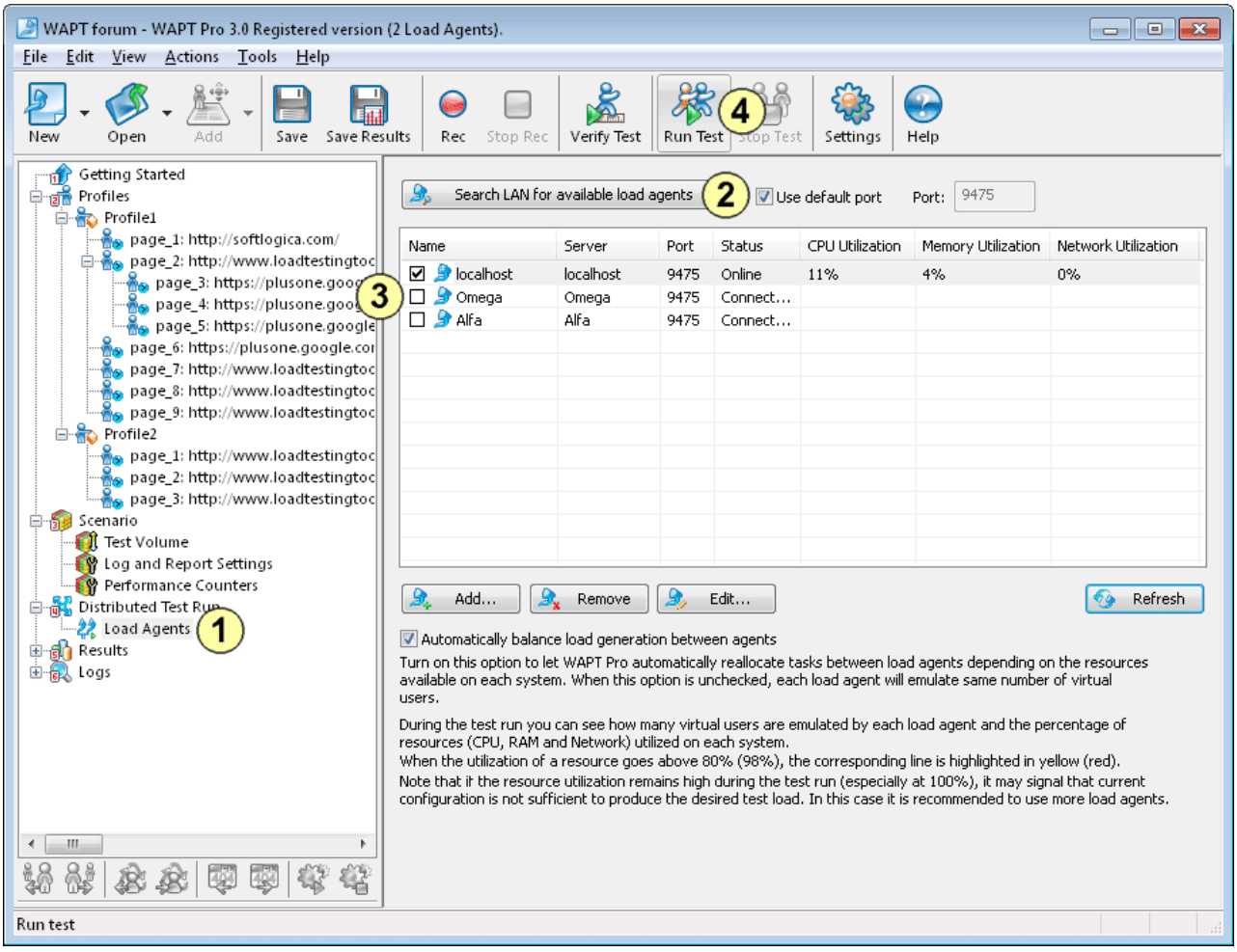
3 Now we have finished designing our test, so we can save it. Click the **"Save"** button on the toolbar to save your test scenario to a file. All profiles will be also saved to the separate files in the same folder. To open the same test in the future you will need to keep all those files.

# Selecting load agents* and starting the test

WAPT Pro allows you to create test load with the help of load agents installed on different computers. The more agents you use, the greater test volume you can create. Test execution is managed from the workplace component and all statistics is gathered from all agents to a single report automatically.

1 Select the **"Load Agents"** item in the left view inside the **"Distributed Test Run"** folder.

2 Click **"Search LAN for available load agents"** to search your Local Area Network for computers with installed agents. If you have agents outside LAN, you can add them manually using the **"Add"** button. Since the connection between the workplace component and load agents is done over TCP/IP, agents can be installed remotely, provided that they are available over the Internet.




3 Put checkmarks near agents that you want to use in the test. You should check at least one.

4 Click the **"Run Test"** button on the toolbar to start your test.

* *Available only in Pro version*

# Test results

You can start monitoring the test results right after you launch the test. You may only need to wait several seconds for the first statistical information to come.

(1) The results are represented in the form of Summary Report, Summary Graphs and graphs for each user profile and single request. You can select the desired option in the left view.

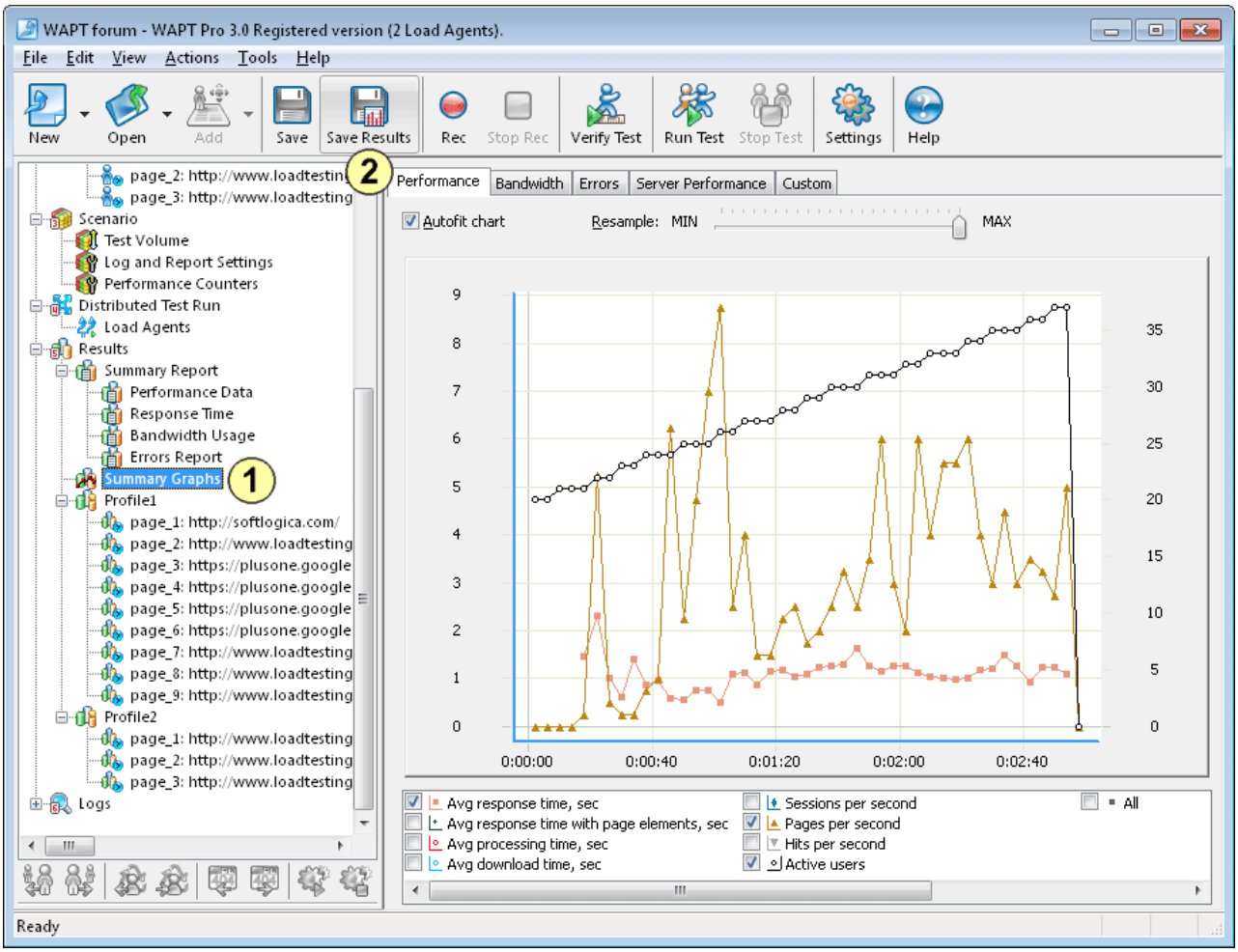
The most important parameters that you can find in the results are described below.

**Error rate** - the number of page requests or user sessions completed with errors as a percentage of the total number of requests or sessions. Errors can be either reported by the server or detected as a result of network problems, wrong server responses and timeouts.

**Response time** - time required by your web site to provide the correct reply to a single page request. It can be measured together with the time required to download all page elements or without it.

**Number of pages per second** - number of page requests successfully served by your web site per second.

To estimate the performance and reliability of your web site you should analyze how the above values change during the test depending on the test load.

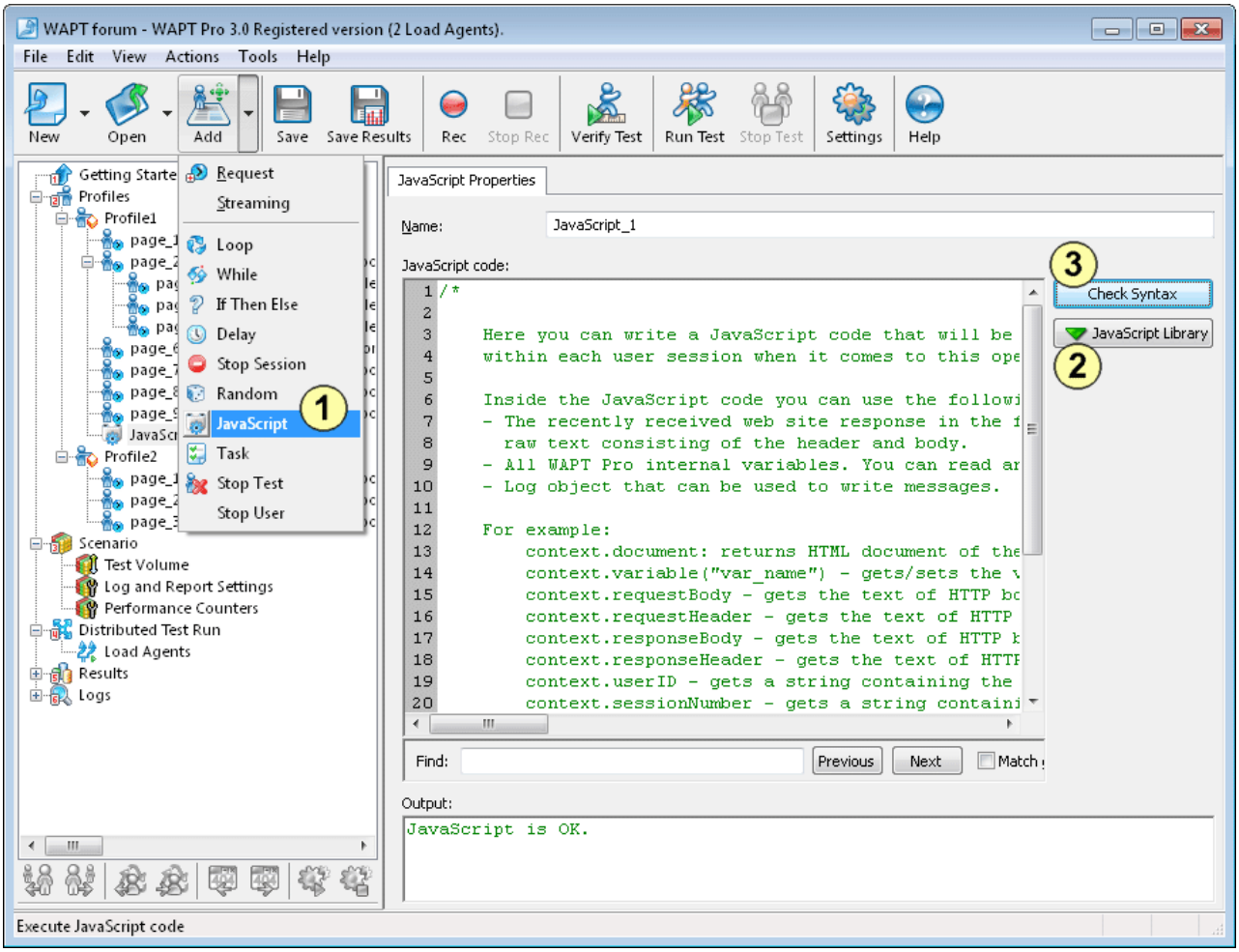



(2) You can save the results of the test run either as a single report in the HTML format, or as a special .wpr file. In the latter case you will be able to open that file with WAPT again.

# The use of JavaScript*

You can include the execution of a JavaScript code into virtual user profiles. This provides you with the most universal way to perform any specific and complex calculations required to set the correct values of the request parameters.

1 Select a request after which you want to execute a JavaScript code. Choose **"Add | JavaScript"** on the toolbar. The JavaScript operator will be added to the profile. Select it to edit the code in the right view. Initially the edit field contains a comment with a short instruction on how to use this feature.




2 In your code you can use functions defined in the WAPT Pro JavaScript library. Click the **"JavaScript Library"** button to extend it by adding more .js files.

3 Click the **"Check Syntax"** button to check the syntax of your JavaScript code. The result of the check will be displayed in the “Output” window.

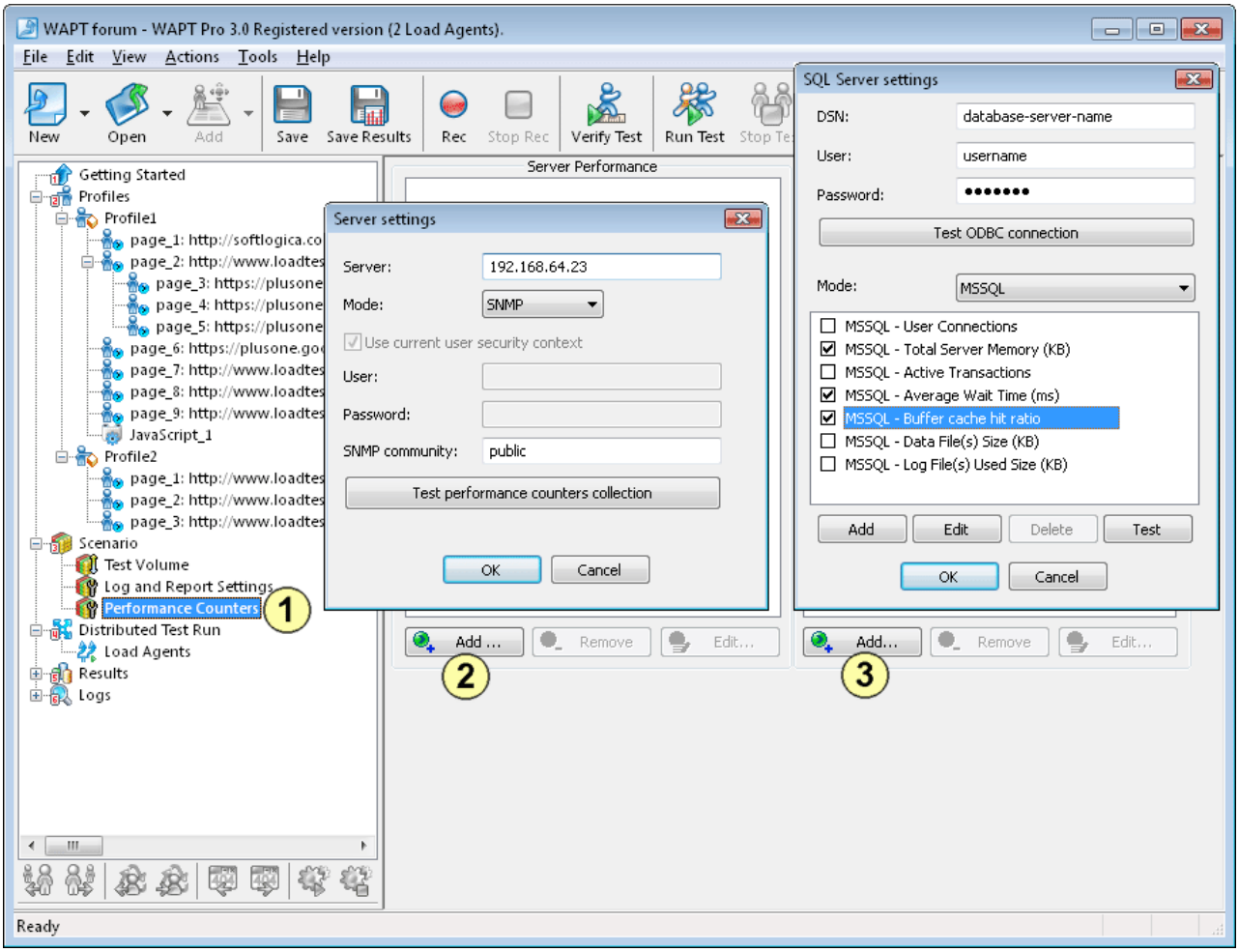
There is another way to use JavaScript code in your profiles. You can call functions defined in the JavaScript library directly when you specify how to calculate values for request parameters and variables. You can do this in the properties of any request the same way you use other internal WAPT Pro functions.

** Available only in Pro version*

# Performance counters*

In addition to the external performance parameters, such as response time, WAPT Pro can collect performance data directly from the server that you test. This data is included in separate tables in the report along with other parameters.

1 Select the **"Performance Counters"** item in the left view.




2 Now in the right view you can add one or several servers that are used to run your web site. WAPT Pro will connect to those servers and collect CPU, disk space, memory and network usage data during the test. You can choose between using WMI and SNMP interface for each server. Select one that is supported by each system and specify other connection options.

3 You can also monitor one or several database servers. This is done over an ODBC connection to each database. WAPT Pro includes a set of predefined counters for MS SQL and Oracle databases, but you can add your own ones that will read any required performance parameters. For each counter you should specify the SQL query that is used to retrieve it from the database.

** Available only in Pro version*