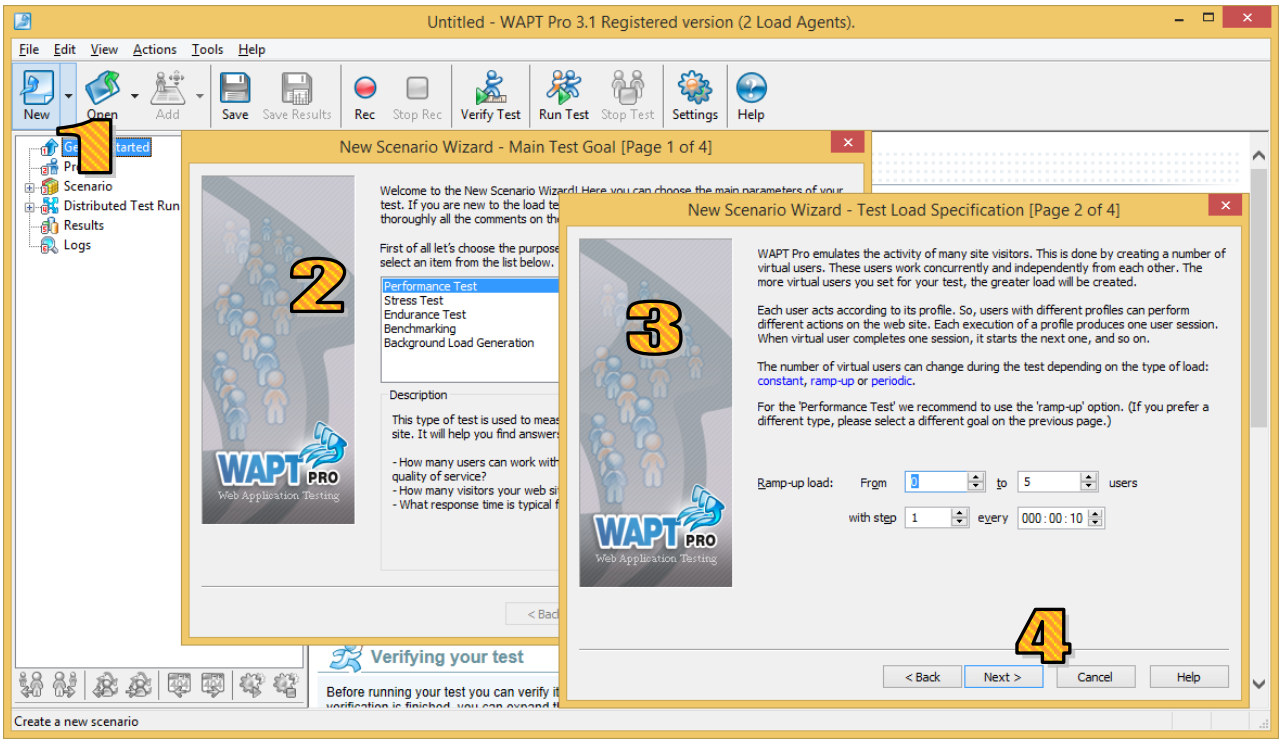
Load testing with **WAPT**: Quick Start Guide

This document describes step by step how to create a simple typical test for a web application, execute it and interpret the results. A brief insight is provided on the additional features available in WAPT Pro, extension modules and the x64 Load Engine.

Creating a test scenario

All the general parameters of a test, such as the number of virtual users, type of load and test duration are specified in the scenario. So, to create a new test you need to create a new scenario.

1 Click the **"New"** button on the toolbar. This will launch the **New Scenario Wizard**.



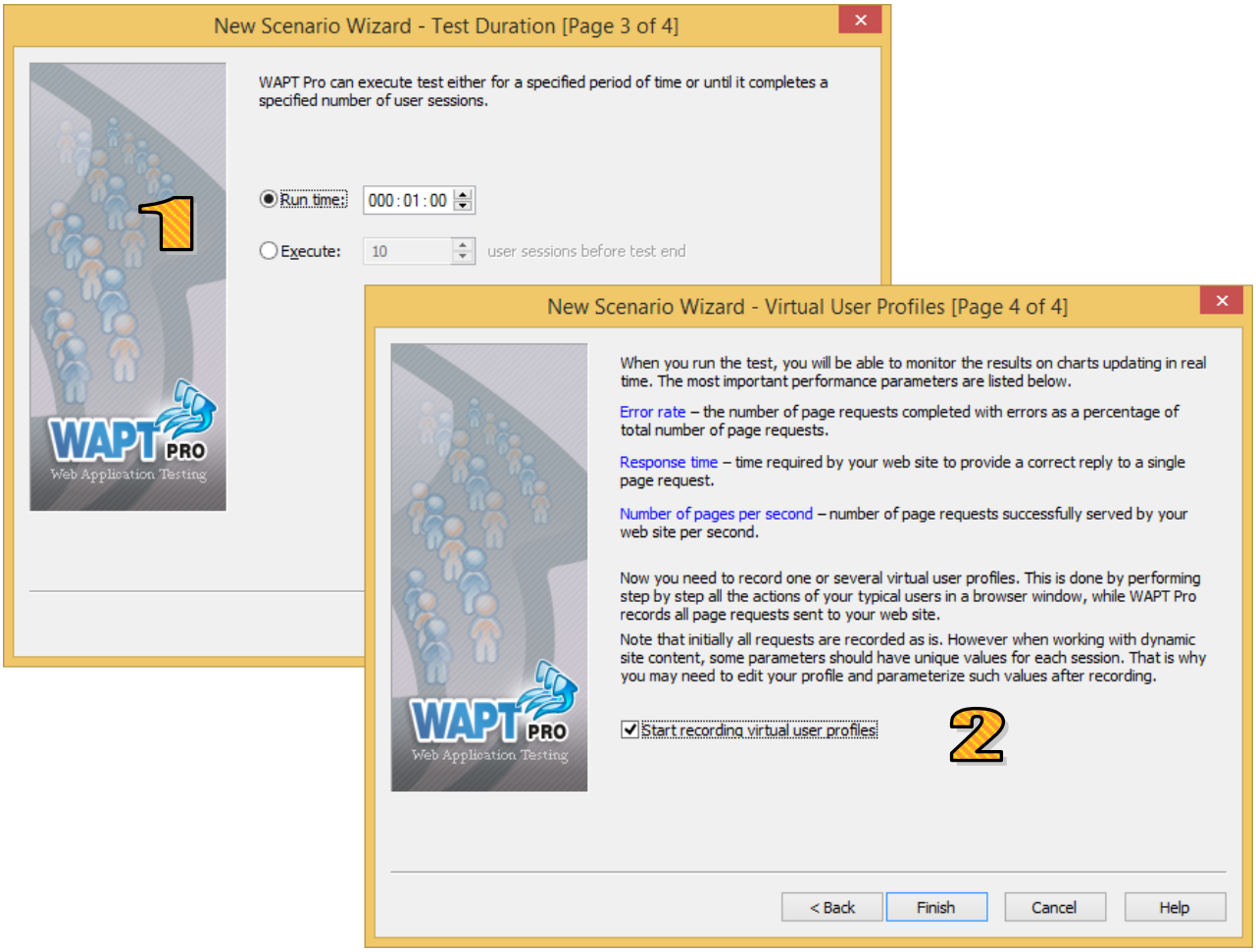
2 In this example we will create a simple performance test. Choose the "Performance Test" option on the first page of the Wizard and click the **"Next"** button to continue.

3 On the second page of the Wizard you can specify the basic parameters for the ramp-up load recommended for the performance testing. The number of virtual users will grow during the test and you can make it grow faster or slower. This will let you compare the performance of your web application on different test phases depending on the changing load.

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

Test duration options

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



2 The last page of the Wizard contains some important hints on how to create a test and interpret its results. Click the **"Finish"** button to proceed to the test recording.

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 design of the virtual user profiles. One profile is created for each type of virtual users. It contains user path through the web site and other parameters required for the correct emulation of the user sessions. One profile is usually executed by multiple virtual users concurrently. You need to create more than one profile only if you expect that some users will have significantly different behavior and/or will visit different parts of the site being tested (like site admins and regular users).

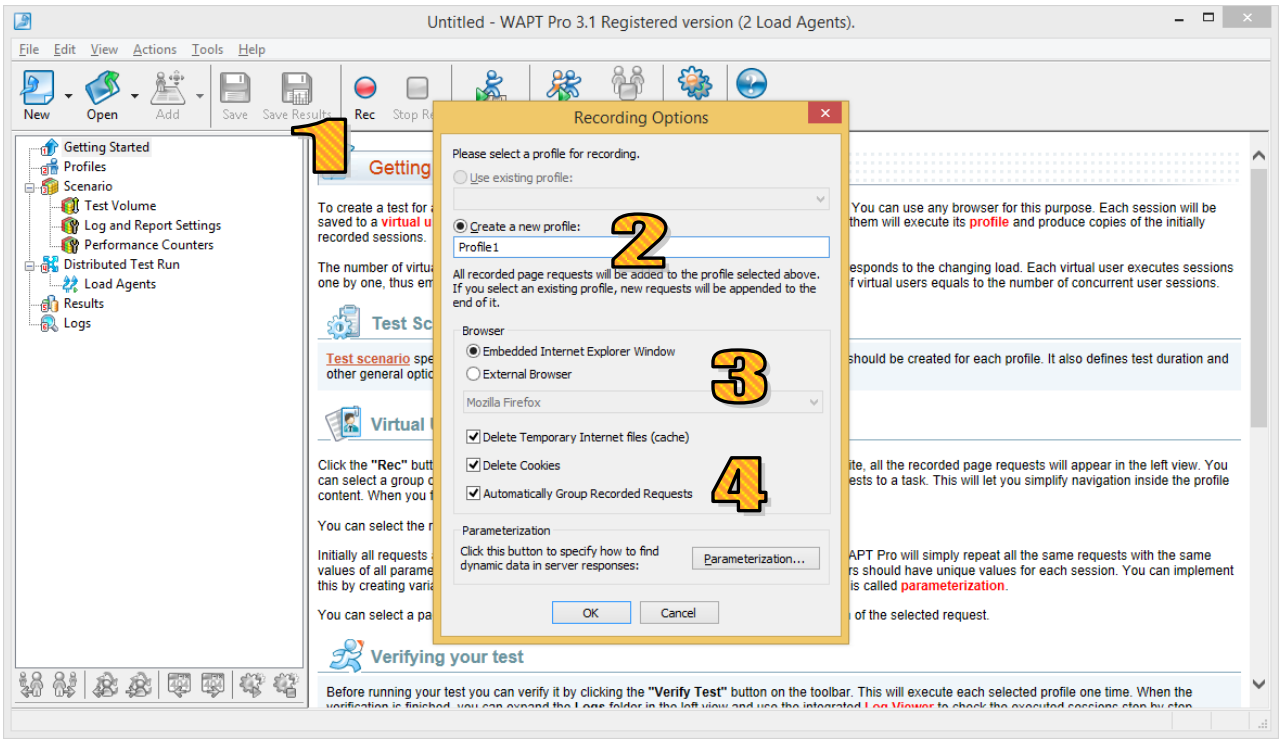
One execution of a profile creates one user session. As soon a user finishes a session, it starts a new one, and so on until the test is finished. So, during the test each virtual user emulates multiple successive real users visiting the site one after another.

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 same sequence of requests with modified parameters. This modification is also called ***"parameterization"***. It is required because some values should be different inside each emulated user session.

1 After you complete the New Scenario Wizard, WAPT will automatically proceed to recording a profile. In the future you can click the **"Rec"** button on the toolbar to record another one. This will open the **"Recording Options"** dialog.

2 Choose a name for your profile.



3 You can either use the embedded Internet Explorer window for recording, or choose an external browser for this purpose. Embedded window 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 or some application features not working), try using one of the external browsers instead.

4 It is strictly recommended to delete browser cache files and cookies before starting the recording. This is required to record and then 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.

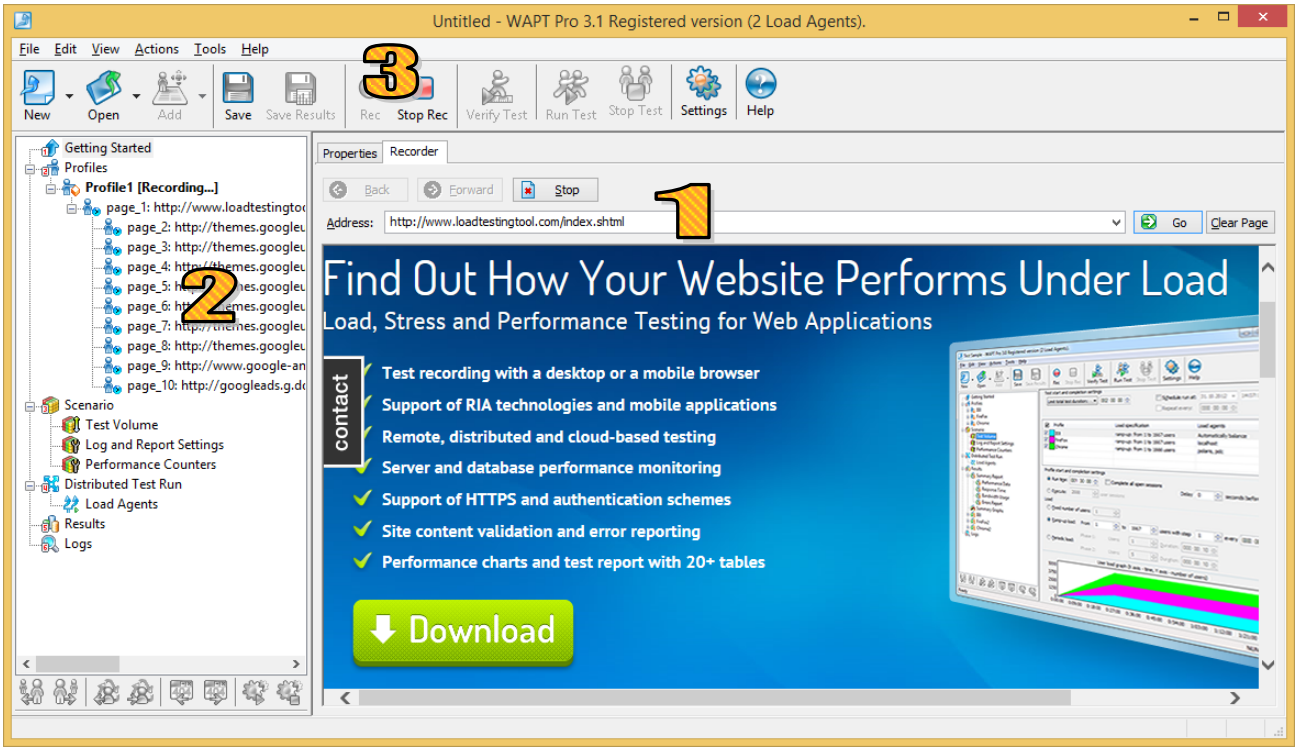
Click the **"Ok"** button to start recording.

Recording a user session

1 Type the URL of your web site or application 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 in 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. Additional requests are initiated by JavaScript code running on the page. They can appear when you work with the web interface controls or at random times. This is also referred to as AJAX.

If you record using the embedded browser, WAPT will place these requests in a tree. In case of an external browser all such requests will be also recorded, but will appear on the same level with other requests. You can organize them into a tree manually.



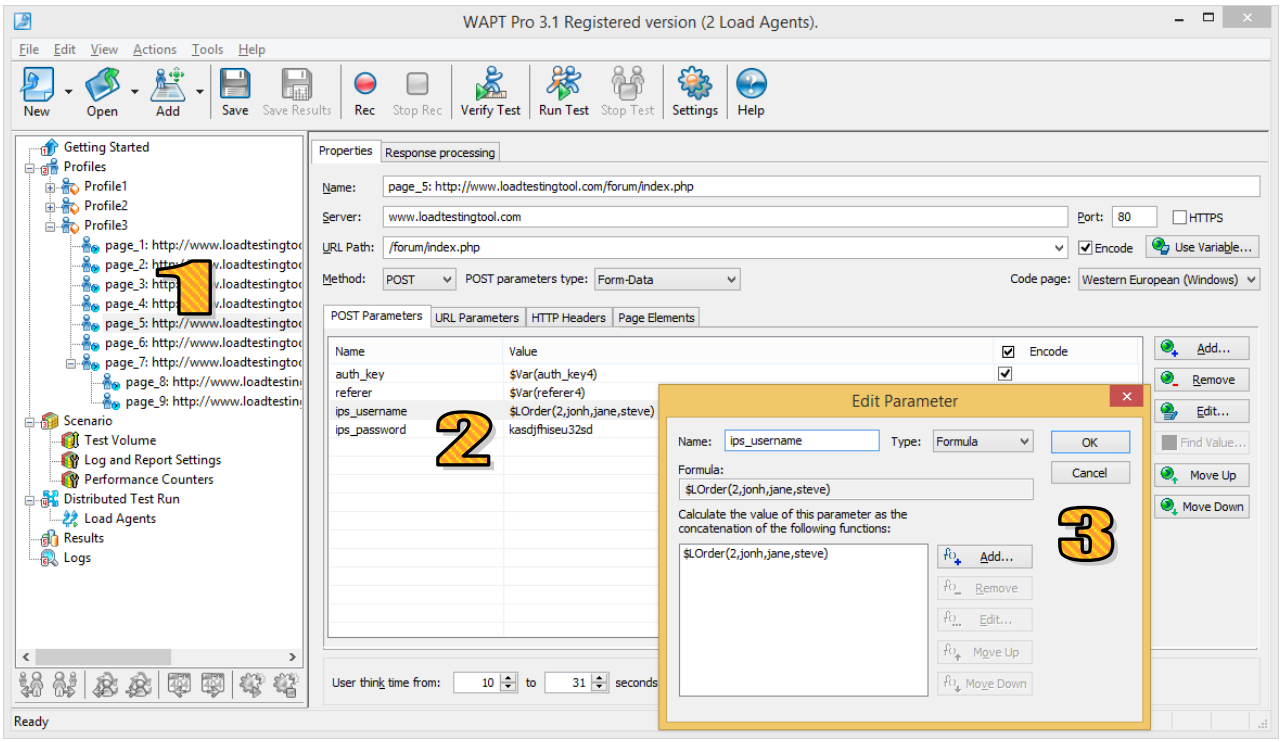
Additional requests to images, .js and .css files do not appear in the list. They are recorded as ***"page elements"***. After recording you can find them in the properties of the corresponding page request.

Note that WAPT works as a proxy server between the browser and the target web site. Because of that you may experience some delays when working with your web application. Try to use it without haste. Do not click next link until you see that the current step has been fully recorded.

3 After you finish recording, click the **"Stop Rec"** button on the toolbar. You can record several more profiles in a similar way, or proceed with just one.

Properties of a request and its parameterization

1 Select any 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 you have to find the right ones manually and modify them according to your needs.

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**. 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 to the right of the list (or just double-click the parameter line).

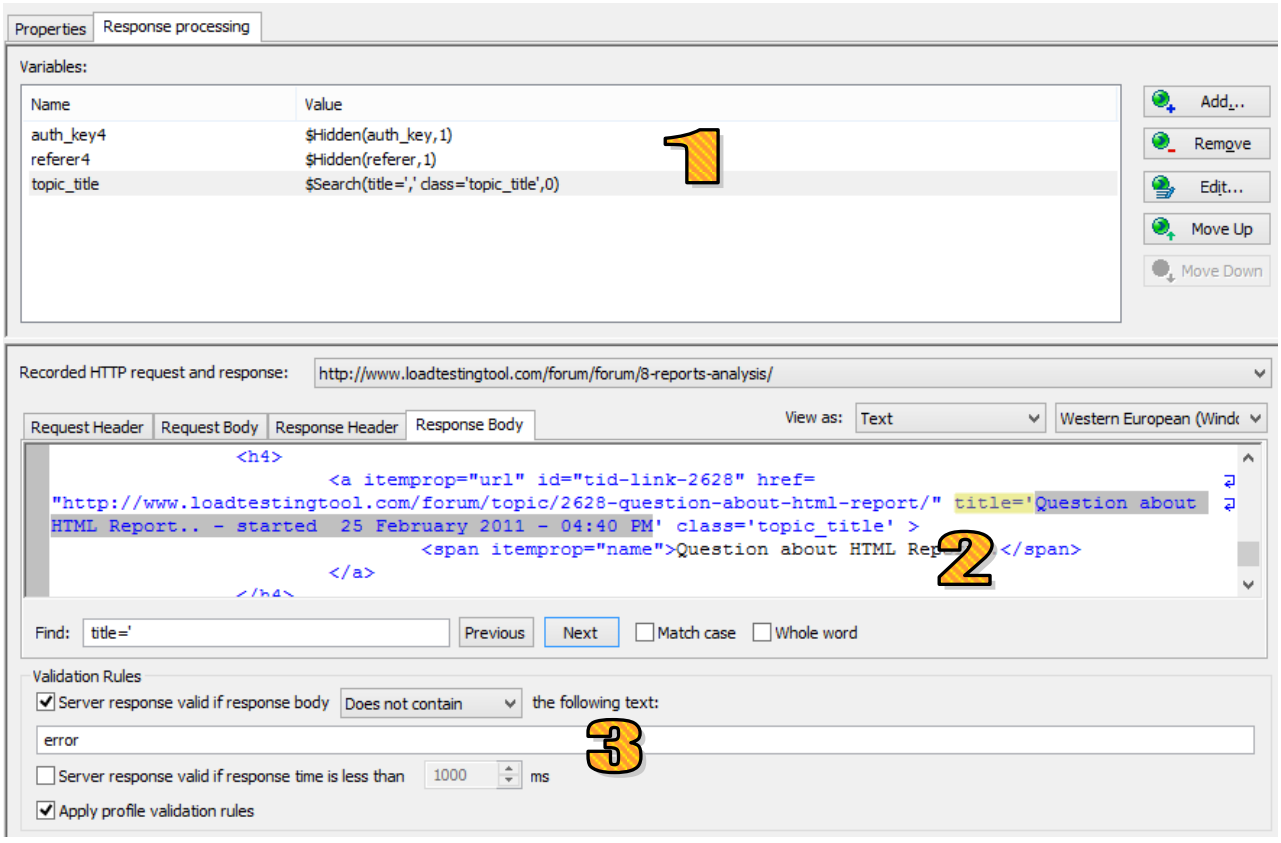
3 In the **"Edit Parameter"** dialog box you can specify how to calculate the value for the selected parameter. In the example shown on the screenshot above the value of the parameter is calculated with help of the **"Ordered List"** function that takes one of three names from the list. You can specify longer list to have more samples, or provide a file with values instead. Other functions can generate random values, extract them from server responses and take them from **variables** assigned earlier.

The processing of server responses

Switch to the **"Response processing"** tab. Here you can specify what WAPT should do with the server response to the selected request in each emulated session.

Let's suppose that the application you need to test works with some items or documents. During each session user selects an item from the list, modifies its properties and saves changes. To emulate such session properly you need to specify correct item ID in the properties of the requests that retrieve the item and save it. However this ID is different in each user session. You can only take it from the list of items returned by the server.

That is why you may need to specify how to extract the required value from the server response and assign it to a variable. This variable can be used in all subsequent requests instead of the initially recorded value.



1 The list of variables is provided at the top of the **"Response processing"** tab. You can assign variables using the same set of functions as for the parameters. The difference is that variables are updated after receiving server response, whereas parameters are calculated before sending the request. In the above example, the **"$Search()"** function is used to extract a random topic title from a page containing several such titles. This is done in a test for a forum web site.

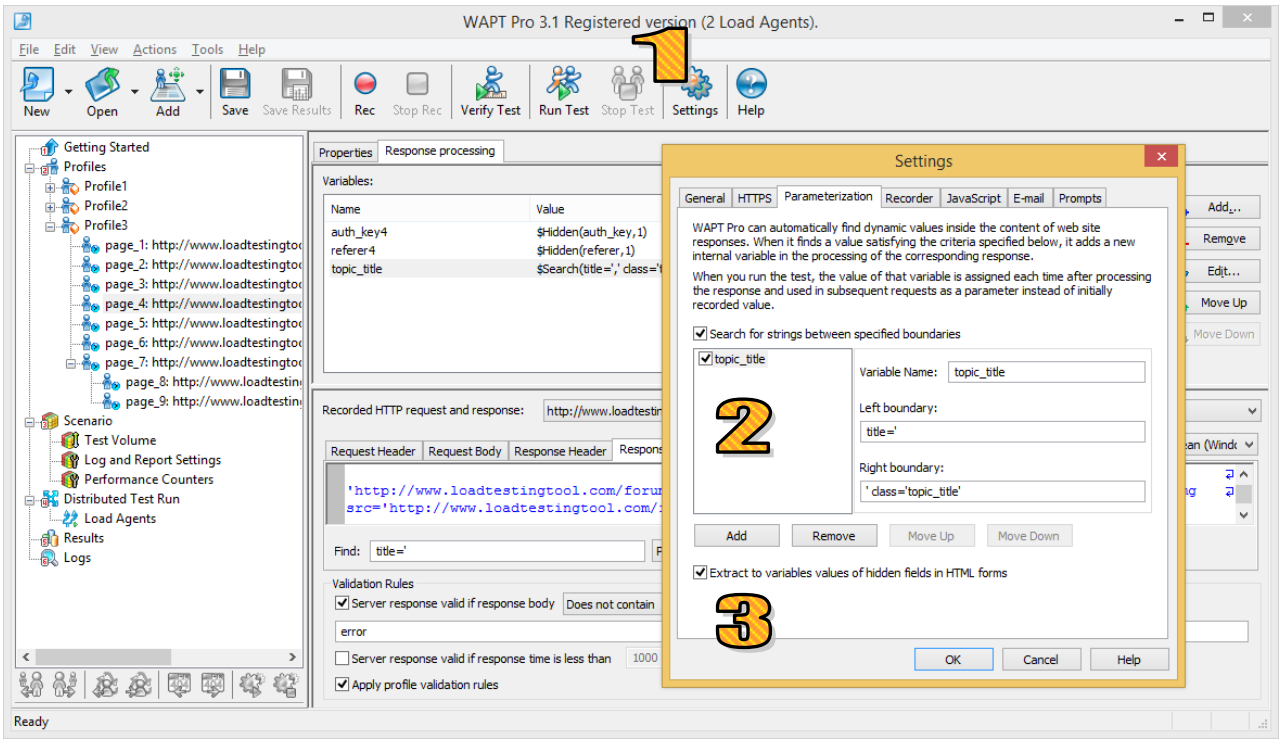
2 The tab contains full information on the initially recorded request and response to it. You can search for any text there. This is very useful, if you want to find the initially recorded value and its bounding text which can be used to specify the arguments of the **"$Search()"** function.

3 There is one more useful option in the processing of server responses. You can specify custom validation rules here. You can make WAPT identify application-specific errors even if they are not reported in the standard way. In the above example the response is treated as valid, if it does not contain the *"error"* word inside.

Automatic parameterization

The parameterization procedure described above can be a rather complex and time consuming task. However if you have performed it for a profile, you can automate this process for any similar profile you create in the future.

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

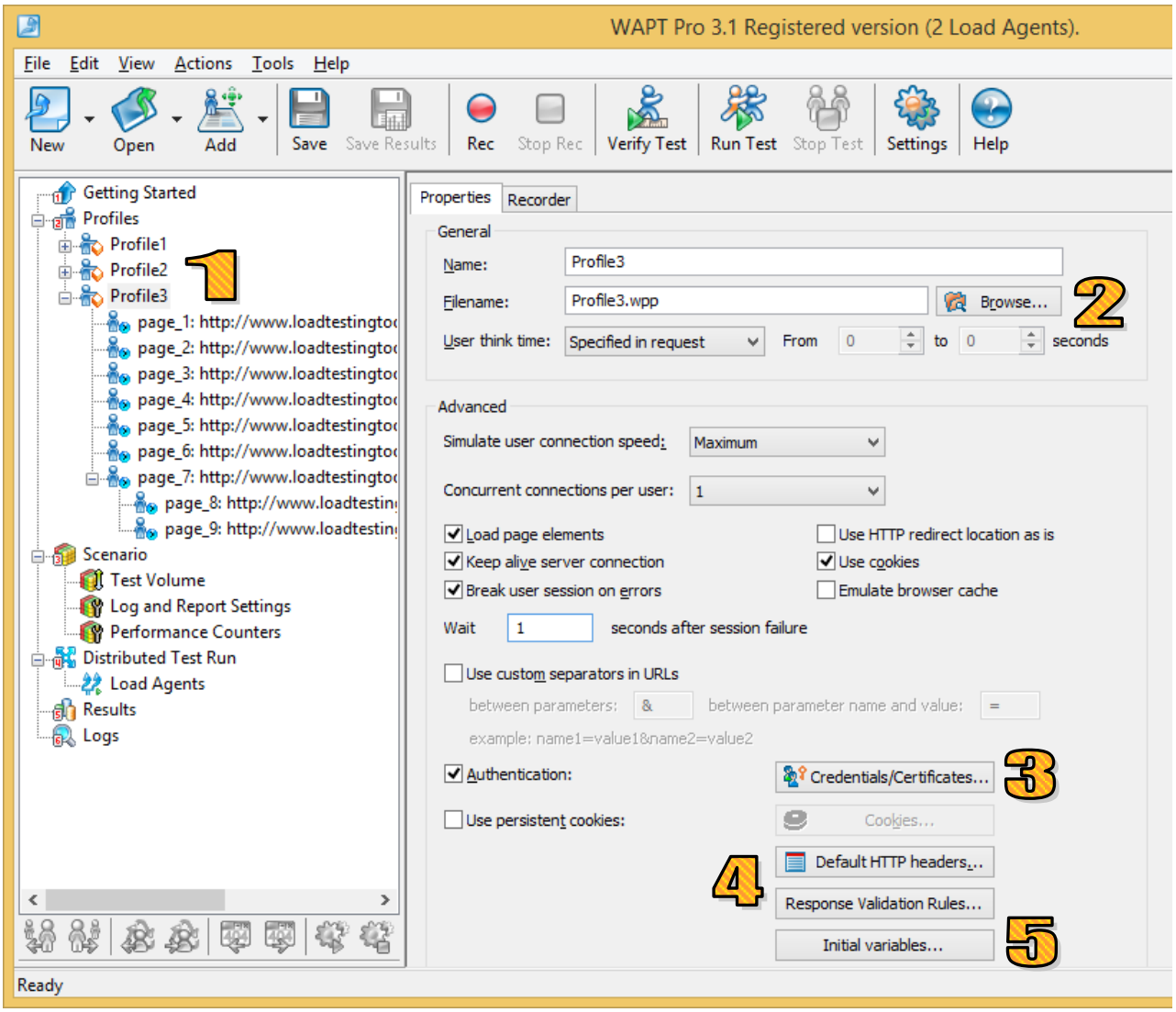


2 You can specify a number of rules for dynamic values contained on the recorded pages. For each value you should specify its left and right boundary. After you finish recording a profile, WAPT will check all the web site responses for such values and will create a new variable for each one. It will also replace each value with the corresponding variable in all parameters of all subsequent requests.

3 The "Extract to variables values of hidden fields in HTML forms" option works in a similar 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 session-specific values from a server to the client. That is why after recording a profile you will probably see many variables created by WAPT and assigned with help of the **"Hidden"** function. You can also see how these variables are used in the parameters of the subsequent requests with help of the **"$Var()"** function. This function returns the value of a variable with a specified name. Since this option is turned on by default, in many cases recorded profiles do not require additional parameterization.

Properties of a virtual user profile

1 There are a number of options associated with each profile. To edit them, select the profile in the left view.



2 Each profile is stored in a file with the **".wpp"** extension. You can change its name here.

3 If your web site requires authentication or a client certificate, check the **"Authentication"** option and click the **"Credentials/Certificates…"** button to specify a set of credentials that virtual users will use when running this profile.

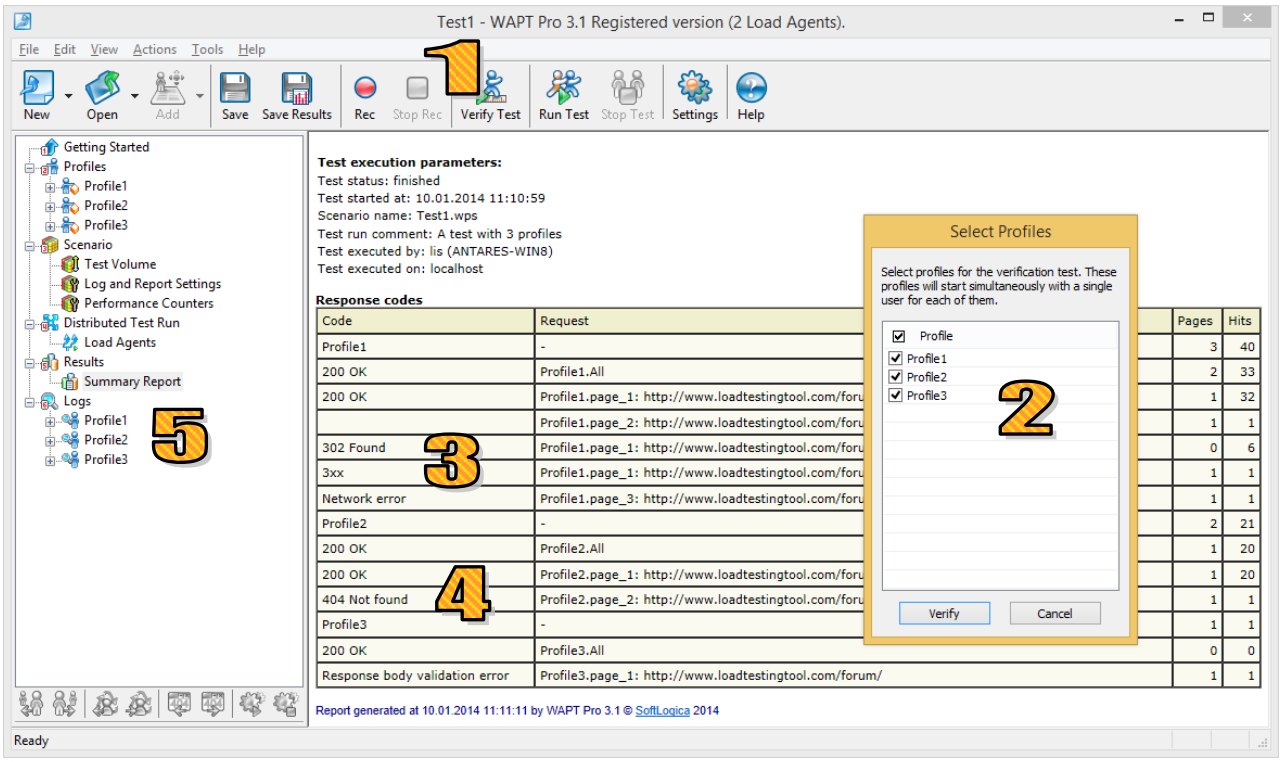
4 Note that **"Default HTTP headers"** and **"Response validation rules"** options can be overwritten in each request.

5 If your profile uses session-specific values in the very first request, you can assign variables before the beginning of a user session. Click the **"Initial variables"** button for this.

Test verification

1 Click the **"Verify Test"** button on the toolbar.

2 WAPT will let you select profiles for verification and will execute each of the selected profiles one time. When the verification is finished, WAPT will show a simple report that contains summary information with the response/status code for each request. This way you can check if the profiles are working correctly and eliminate any parameterization issues before running the actual test.



3 Note that response codes starting with "3" (like 302) are not errors. These are HTTP redirects that are processed by WAPT automatically. Similarly, if you see that some request completed with the 401 code, this is not a problem. This only means that the server requires authentication, so you should provide user name and password in the profile properties. After that the same request will still produce 401 code, but it will be followed with the"200 OK" code. The full sequence of requests is provided in the logs. In the above report you will only see the same request listed for both codes.

4 If you see the 404 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. If you find same problem there, your site contains a broken link.

If you see a "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 get other types of status codes like "Response body validation error" or "Timeout" here.

Note that the table also includes information about responses to page element requests. For this reason it may list same request with different codes. For example, if a page request completed successfully, but its page element was not found on the server, it will be listed as 200 and 404.

5 If any issues are found on verification, you can expand the **"Logs"** folder in the left view to get more information on each problem.

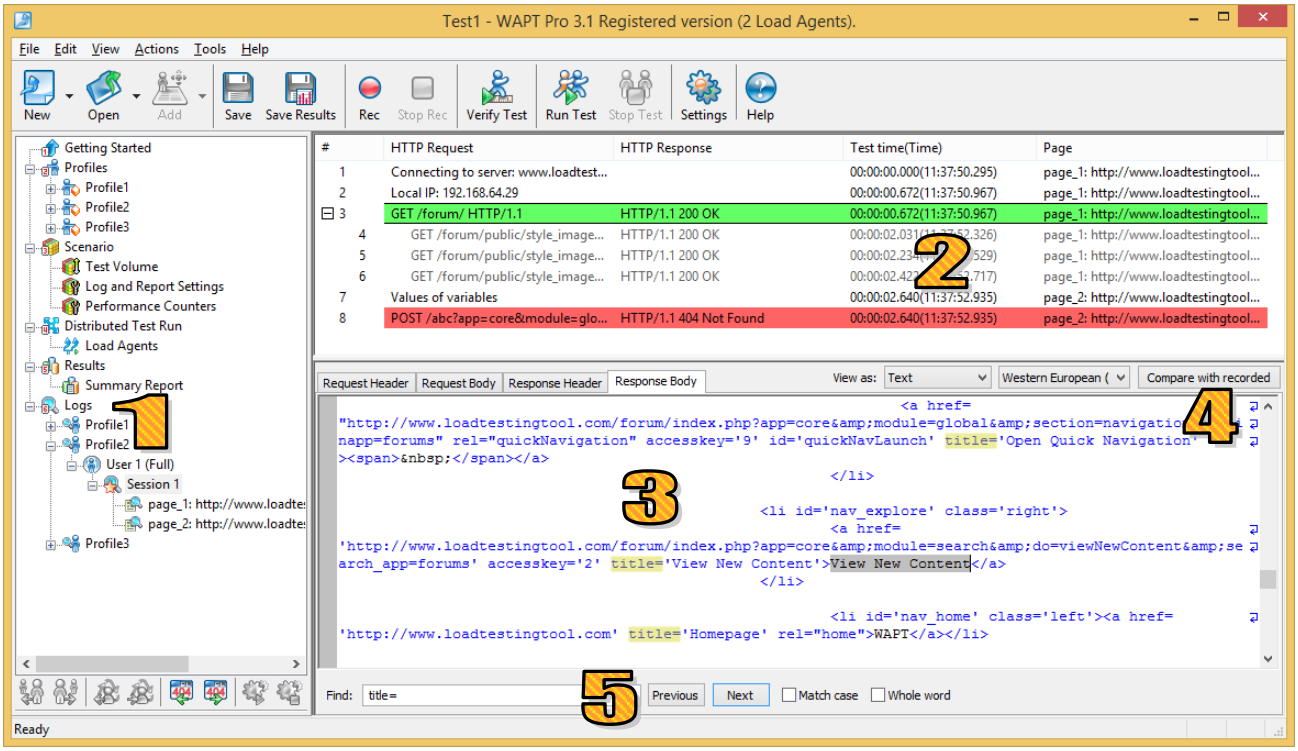
Log viewer

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

1 Expand the **"Logs"** folder in the left view and select a session.

2 In the upper right view you will see the log lines of different types.

- Information messages like "Connecting to…", "Local IP…" and any messages written to the log by JavaScript operators.
- Page requests. Successful ones are painted green. Requests completed with errors are painted red.
- Requests to page elements in grey color with indent under each page request. You can expand and collapse them.
- "Values of variables" lines that provide information on the values of all variables used in the next request.



3 You can select any line and see the details in the lower part of the view. For each request you can switch between different tabs containing request and response headers and bodies.

4 You can compare any part to the initially recorded content. This way you can check what parameter or header values were different in this specific session. You can also see if server responded as usual or produced a significantly different content, which may indicate a problem.

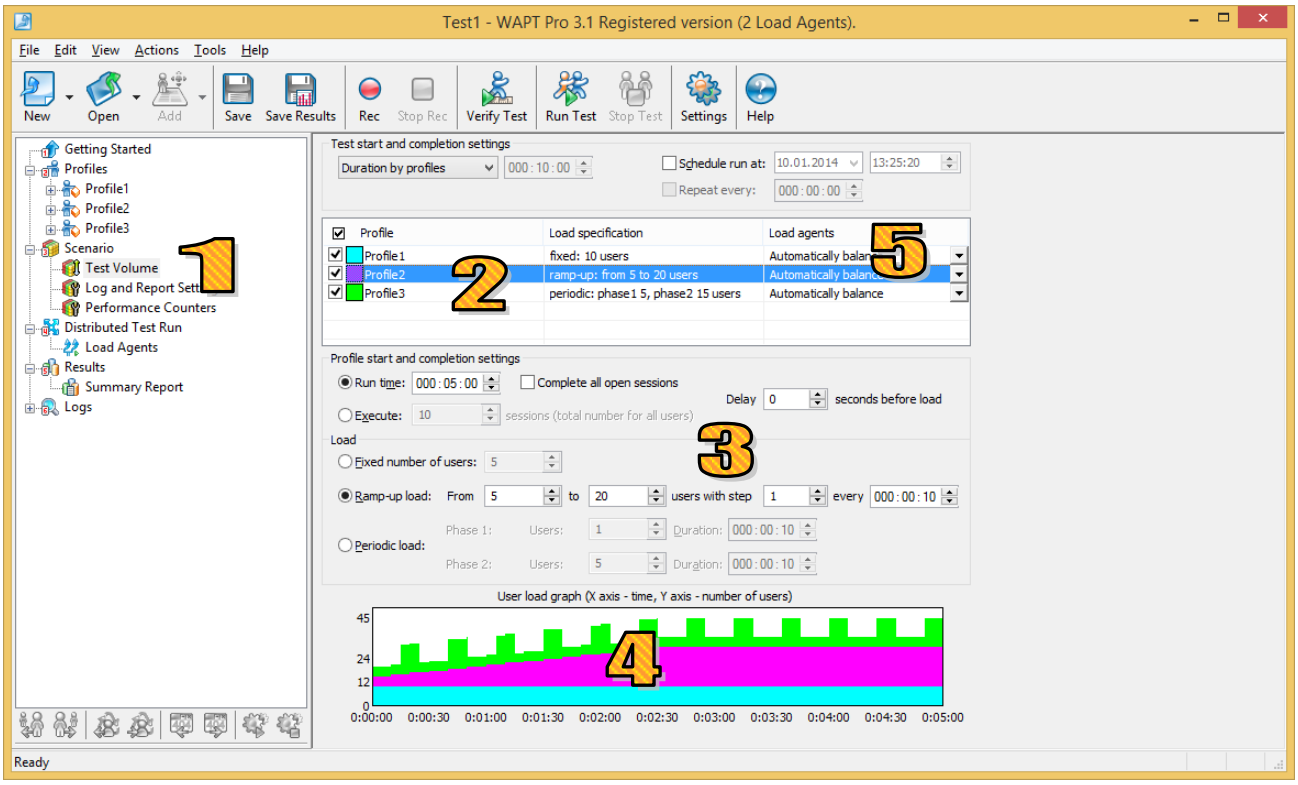
5 The useful search option is also available here.

Note that by default logging is disabled for efficiency reasons. So if you want to get logs after a test run, you should enable this feature on the **"Log and Report Settings"** page. You can save all log files by choosing **"File | Save Logs…"** from the menu.

Test Volume

After you make sure that all your profiles are working correctly you can specify the load parameters for the actual test.

1 Select the **"Test Volume"** item in the left view inside the **"Scenario"** folder.



2 In the right view you can see the list of all your profiles. Check the ones you want to use in the test.

3 You can specify certain load options separately for each profile. Note that these options are shown for the currently selected profile (highlighted with blue selection). If you want to change options of a different one, select it in the list.

In the above example, we have 3 profiles with different types of load (constant, growing and periodic). Second profile (with the rump-up load) is selected and its options are shown below the list.

4 The graph at the bottom of the page shows how the load will be distributed between profiles during the test. Each profile is shown with a different color.

5 In the Pro version of the product you can also specify different load agents for different profiles. By default they are distributed automatically, which means that all agents execute all profiles.

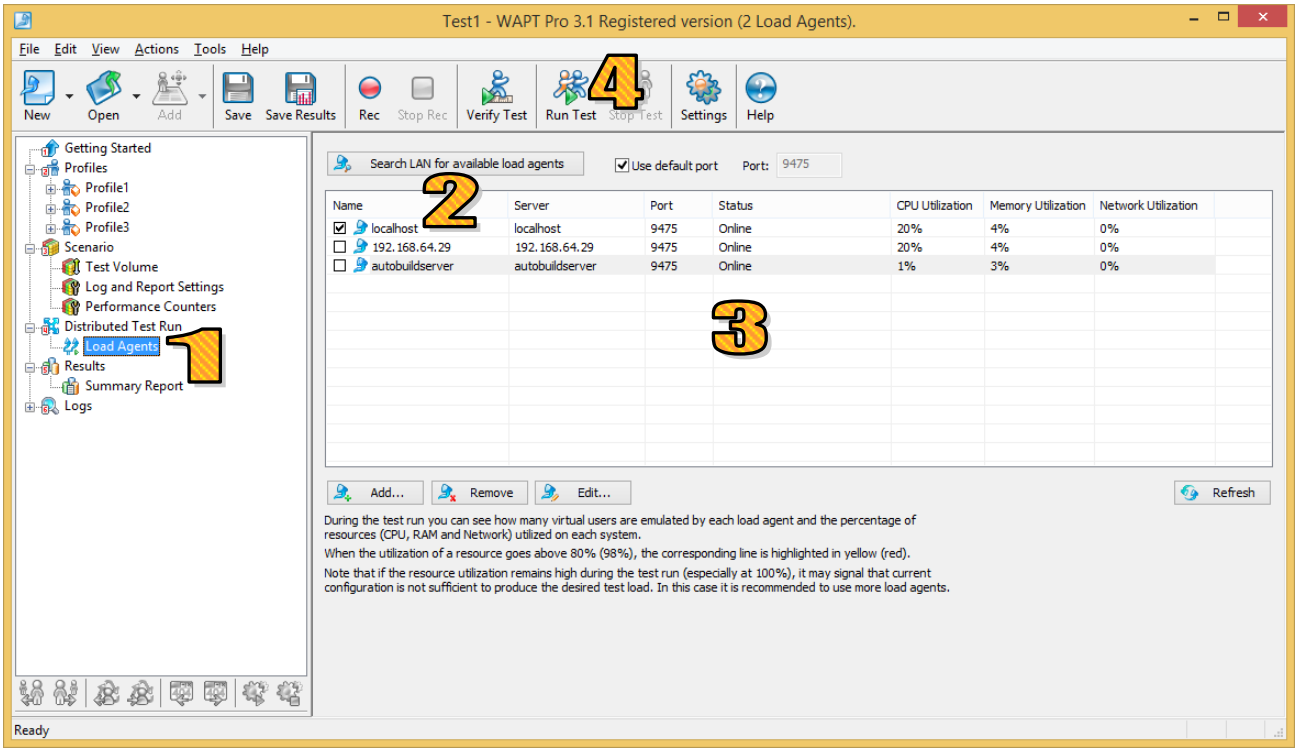
Now we have finished designing our test, so we can save it. Click the **"Save"** button on the toolbar to save your test scenario to a file. All profiles will be also saved to separate files in the same folder. Keep all these files, if you want to open the same test in the future.

Selecting load agents* and starting the test

If you use the Pro version of WAPT, you can generate the test load with help of several computers. Each of them should run a special service called ***Load Agent***. By default one of the agents is installed on the same system with the workplace component of the product. So you can run the test using your own system. However if you need a greater load, you can attach additional load agents.

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

2 Click the **"Search LAN for available load agents"** button to search your Local Area Network for computers with installed agents. If you have remote agents located outside the 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, you need to make sure that your network configuration allows connections to the specified addresses.



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.

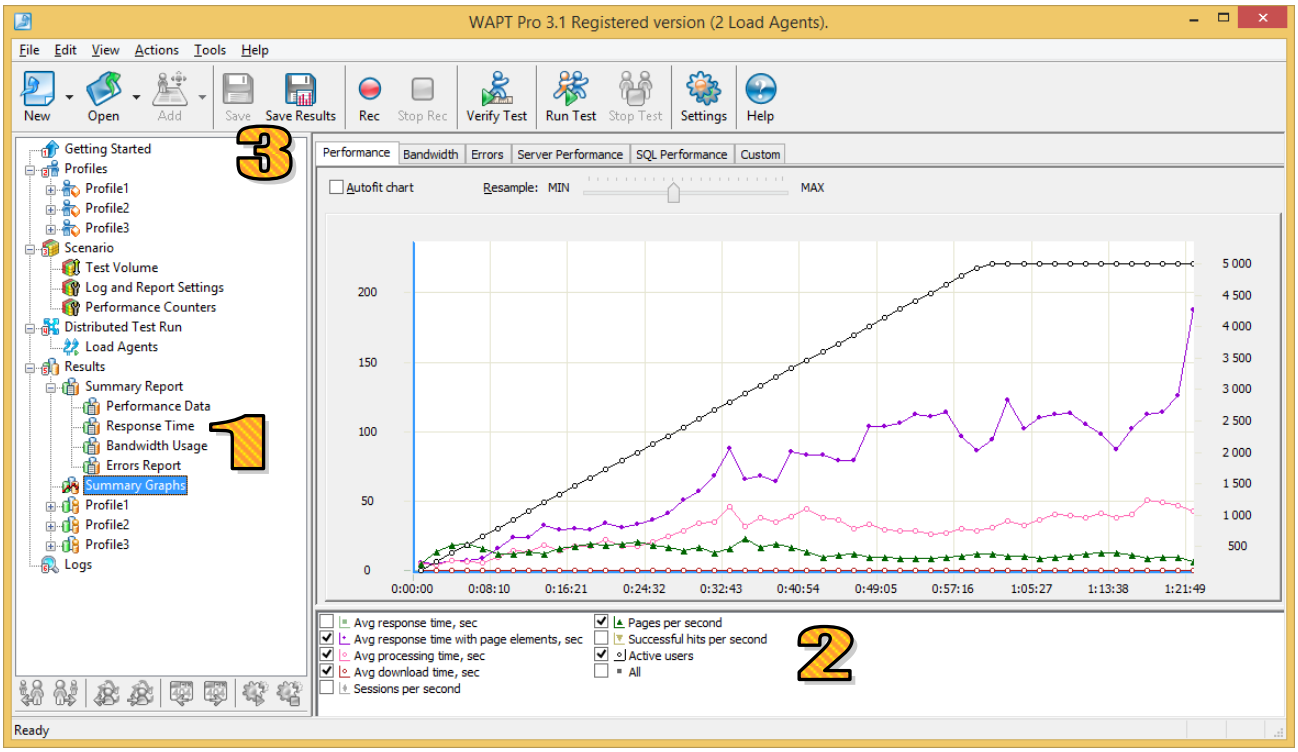
If you use the regular version of WAPT, it uses only one built-in agent, so you do not need to select it. Just click the **"Run Test"** button to start the test.

** 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 data 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 corresponding option in the left view.



2 On graphs you can choose between several tabs at the top and select parameters you would like to see on the graph at the bottom line. Each parameter is shown with a specific shape and color. All graphs have two vertical scales to represent parameter values. Bottom left corner image ( ) near the parameter description means that the value is specified on the left scale. Bottom right corner image ( ) refers to the right scale.

In the above example, the number of pages per second is specified on the left scale, whereas the number of active users is specified on the right scale.

3 You can save the results of a test run either as an HTML report, or as a special results file with the **".wpr"** extension. In the latter case you will be able to open that file with WAPT at any time again, browse graphs and work with the results like right after the test.

Extension Modules

WAPT can test any web application that uses HTTP(s) protocol for client-server communication. It is not usually important which framework and technologies were used to create the application and what hardware and software is used to run its components.

Still there are two situations in which you may benefit from using extension modules that provide support for some specific technologies.

The following modules facilitate the test design process.

- **Module for ASP.net Testing**
- **Modules for JSON Format**

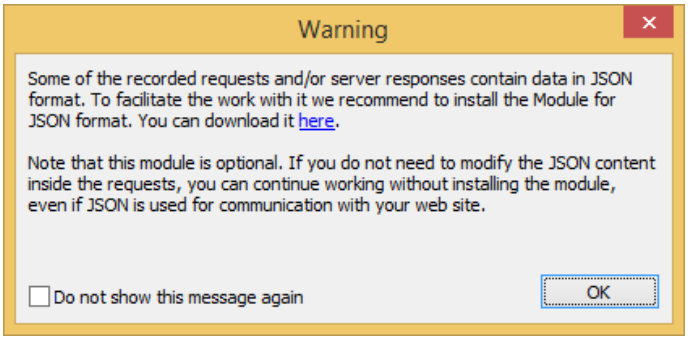
With help of these modules you will be able to create your tests faster and in a more convenient environment. Both modules are optional, but we recommend installing them in case your web application uses the corresponding technology.

Some web technologies allow applications to pass data between the client and the server parts using complex technology-specific structures. To perform conversion of such structures to and from standard XML format, the following modules may be required.

- **Module for Adobe Flash Testing**
- **Module for Silverlight Testing**
- **Module for GWT Testing**

With help of these modules you will be able to parameterize any session-specific values appearing in the test.

It is very easy to find out if you need a specific module to test your web application. When you finish recording a test with WAPT, it will show you the corresponding warning message with recommendations.

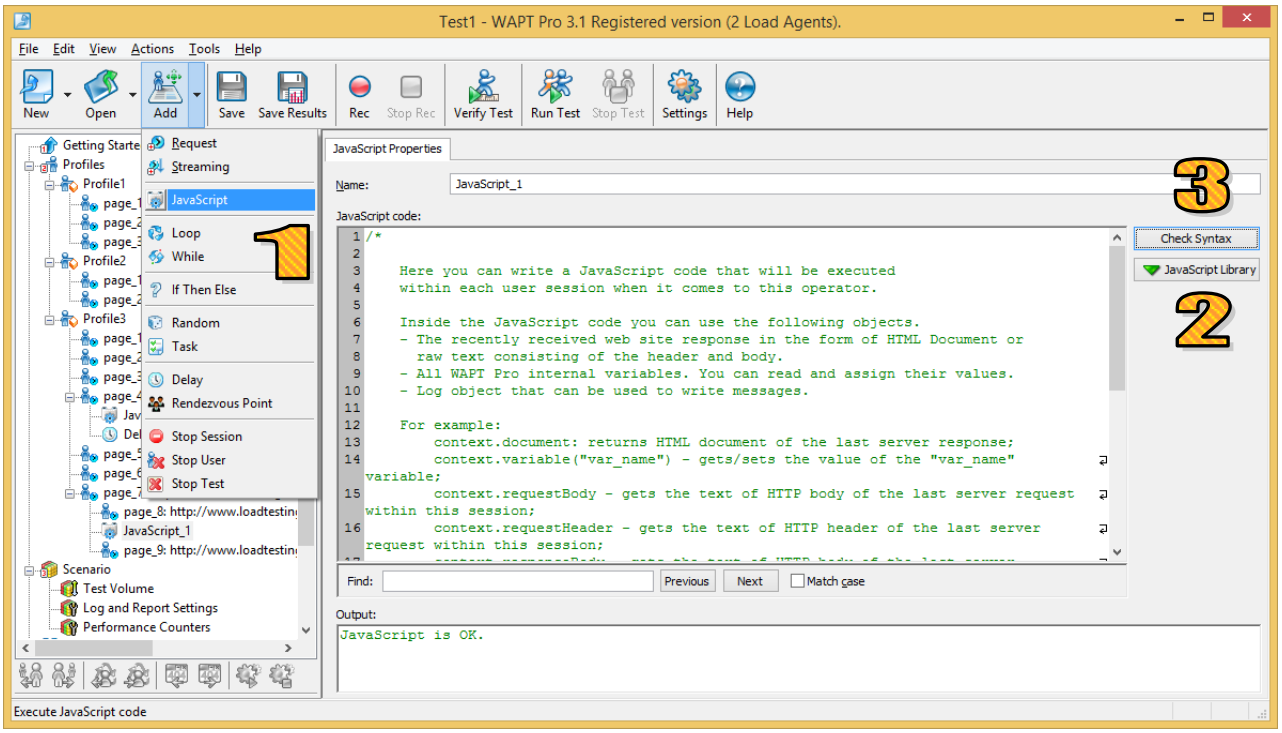


Each of the modules is provided with a separate installation package, but the integration is very smooth. You only need to install the module to the same system where you run WAPT or WAPT Pro.

The use of JavaScript*

In WAPT Pro you can use JavaScript code to perform any calculations inside user sessions. This may be necessary to parameterize session-specific values that cannot be extracted from server responses with help of standard functions available in WAPT.

1 Select the request after which you want to insert 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 window 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 files.

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

Note that JavaScript code is used only to calculate values and assign results to some variables that you will be able to use in subsequent requests. You cannot initiate new requests directly from the code.

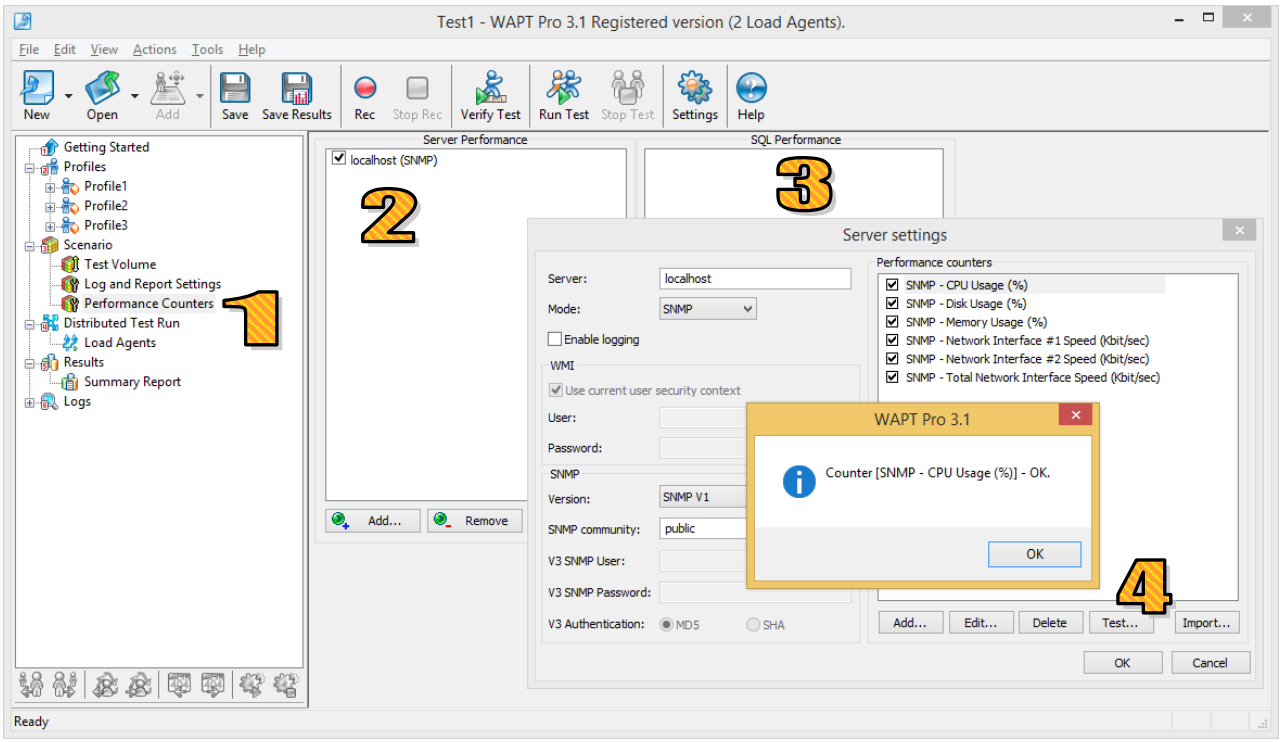
There is another way to use JavaScript in your profiles. You can call functions defined in the JavaScript library directly when you specify how to calculate values for the request parameters and variables. You can do this with help of a special internal function called "JavaScript". It takes the actual name of the function you want to call as an argument.

** 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 information is included in the special tables in the report along with other parameters. You can also see it on graphs.

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 select WMI or SNMP interface for each server.

The value of each counter is retrieved with help of a JavaScript function associated with the counter. You can edit this code and create your own custom counters for any specific values provided by your servers through the WMI and SNMP interfaces.

3 You can also monitor one or several database servers. This is done with the help of an ODBC connection that you need to setup on the system running WAPT Pro. In the SQL server properties in WAPT Pro you simply specify the DSN of the created connection.

A set of predefined counters is provided for MS SQL, MySQL and Oracle. You can also add your own counters for these and different databases. For each counter you need to specify a JavaScript function that will retrieve the counter value using the corresponding SQL statements. You can use default counters as examples.

4 After specifying all connection options you can select a counter in the list and click the **"Test"** button to check that it is working properly.

** Available only in Pro version*

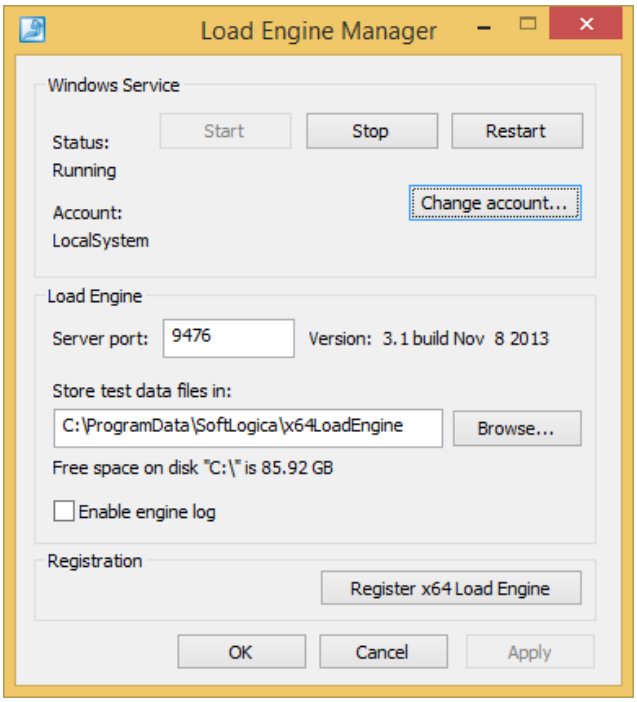
*x64 Load Engine**

Regular load agent that comes with WAPT Pro by default is capable of creating about 2,000 virtual users in an average test. This means that to create 10,000 users you would need to use 5 such agents installed on different systems. In most cases it is preferable to use **x64 Load Engine** instead.

The functionally of the x64 Load Engine is identical to the functionality of the regular Load Agent. A much higher performance is achieved because of its native 64 bit architecture. You can setup the engines on several systems and use them concurrently to create greater test volume. The workplace component of WAPT Pro manages the work of the engines same way it does it with the load agents.

The x64 Load Engine can be installed on any system with 64 bit Windows OS starting from Windows XP. This product is not included in the standard WAPT Pro installation package and should be downloaded separately.

The engine runs as a system service. It can be launched and managed from the **Load Engine Manager** available under the Start menu on the system where the engine is installed. By default this service is launched automatically, so you can start using it immediately after the installation. There is no need to restart it manually after the system reboot.



** Compatible only with Pro version*