Load testing with

WAPT Cloud

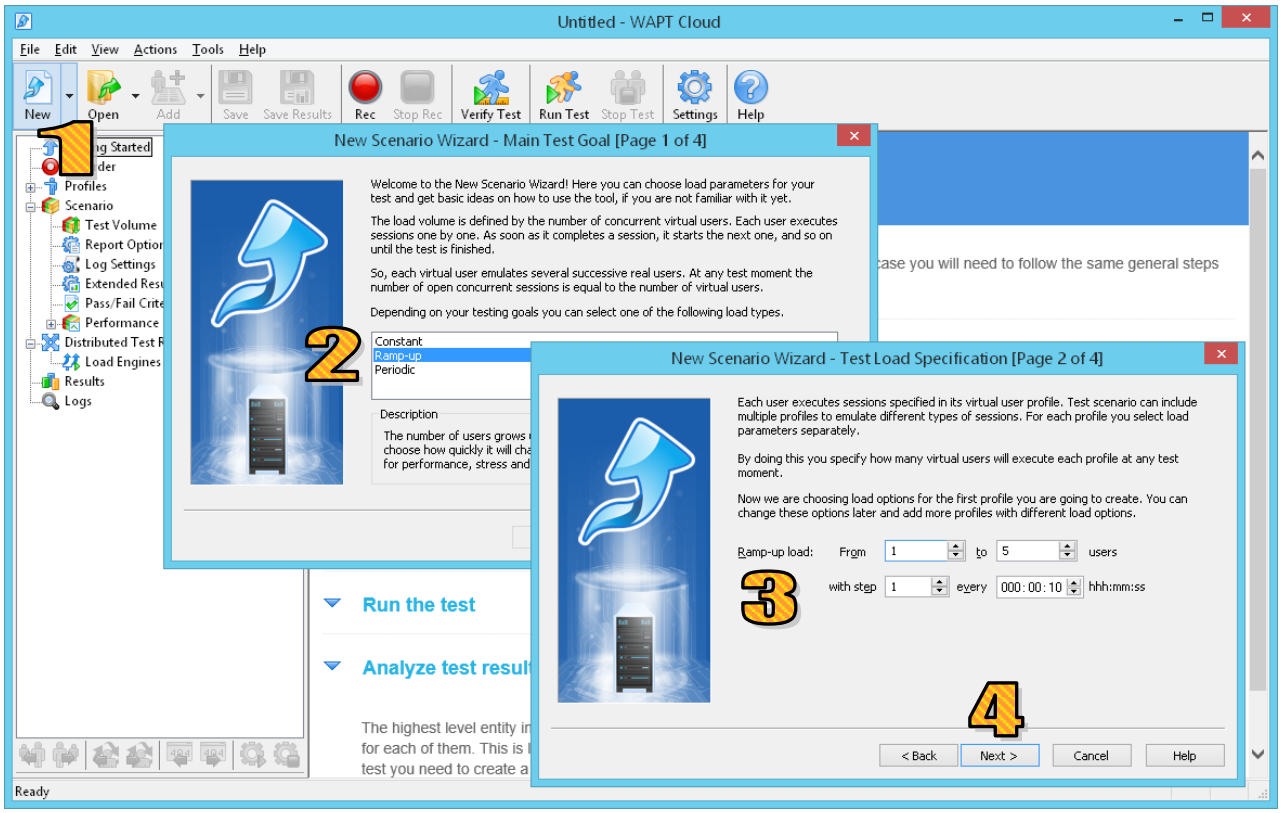
Quick Start Guide

This document describes step by step how to create a simple typical test for a web application, execute it and interpret the results.

Creating a test scenario

We will start with creating a test scenario. It includes all the general parameters of the test, such as the number of virtual users, type of load and test duration.

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



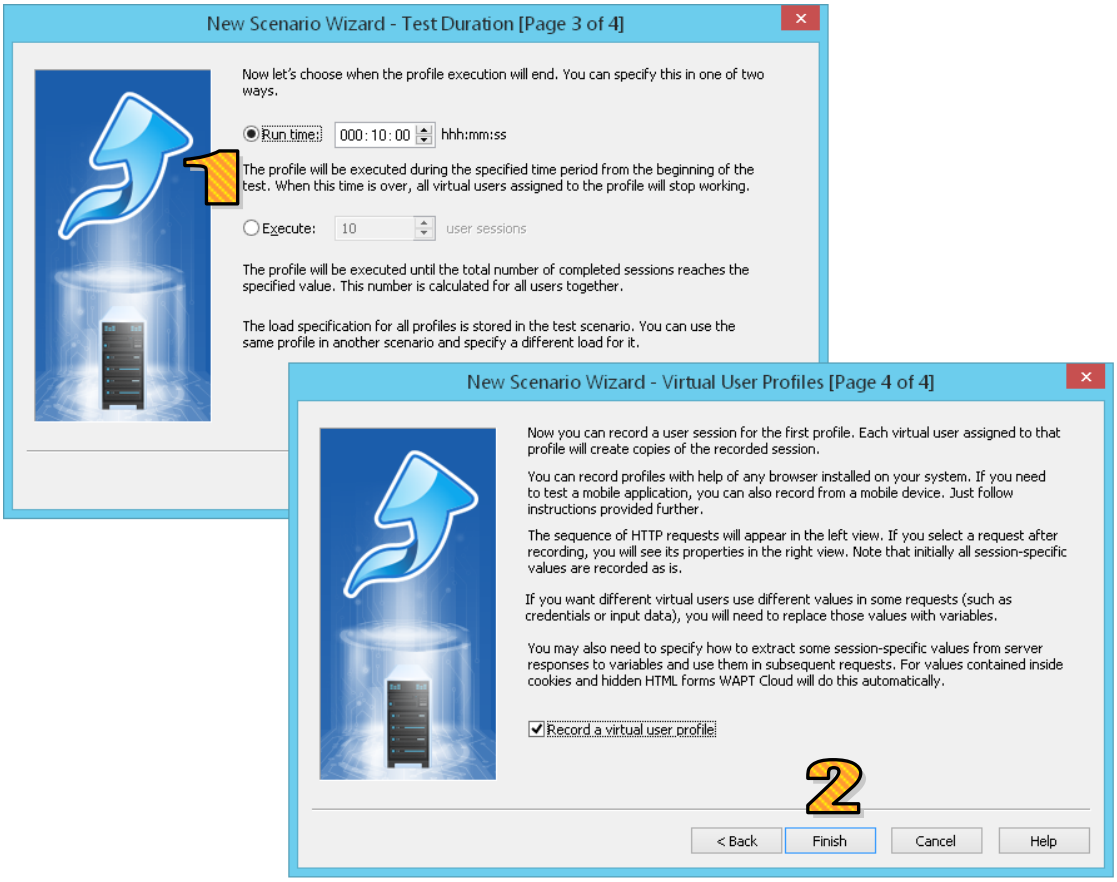
2 In this example we will create a simple performance test. Choose the "Ramp-up" option on the first page of the Wizard and click the **"Next"** button to continue.

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 in different test periods 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 the 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 adjusted later. To do this click the **"Test Volume"** item in the left view of the WAPT Cloud window. The type of test only changes the default values of the load parameters you see in the wizard.

* * *

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).

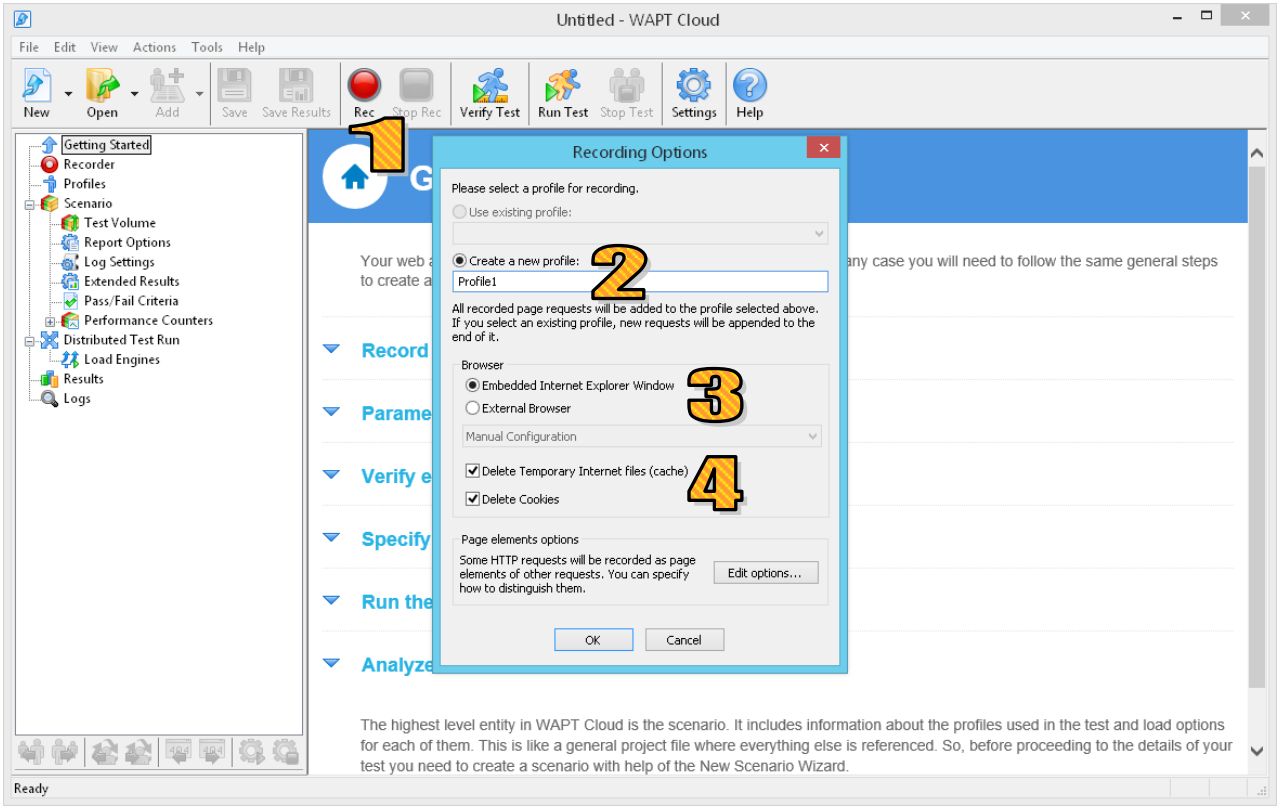
One execution of a profile creates one user session. As soon a user finishes its current session, it starts a new one, and so on until the test is finished. So, during the test each virtual user emulates multiple successive real users visiting the site one after another.

Creating a virtual user profile

Initially profiles are recorded with the help of a browser. You should simply perform step by step all the actions of the user that you want to emulate. WAPT Cloud will record the produced HTTP requests. During the test WAPT Cloud will execute multiple copies 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 parameters should have different values in each emulated user session.

1 After you complete the New Scenario Wizard, WAPT Cloud 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 Cloud better organize the recorded requests, but it has limited JavaScript support. So, 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 a session independent from all the previous user activity that may have been performed on your system. Only in such case it will be possible to reproduce the recorded session correctly. WAPT Cloud will perform the cleanup automatically if you leave the corresponding options checked. Note that it may take up to several minutes when you do this for the first time.

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 Cloud will record all the HTTP requests generated by the browser.

2 You will see the requests appearing in the left view inside the **"Recorder"** folder. Initially they are recorded as a simple list without any structure. All images and AJAX calls are placed on the same level with the pages of your site. You cannot edit the requests while recording, but you can add bookmarks. This is useful to mark each separate action inside the user session.

Sometimes when you click a link WAPT Cloud 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.

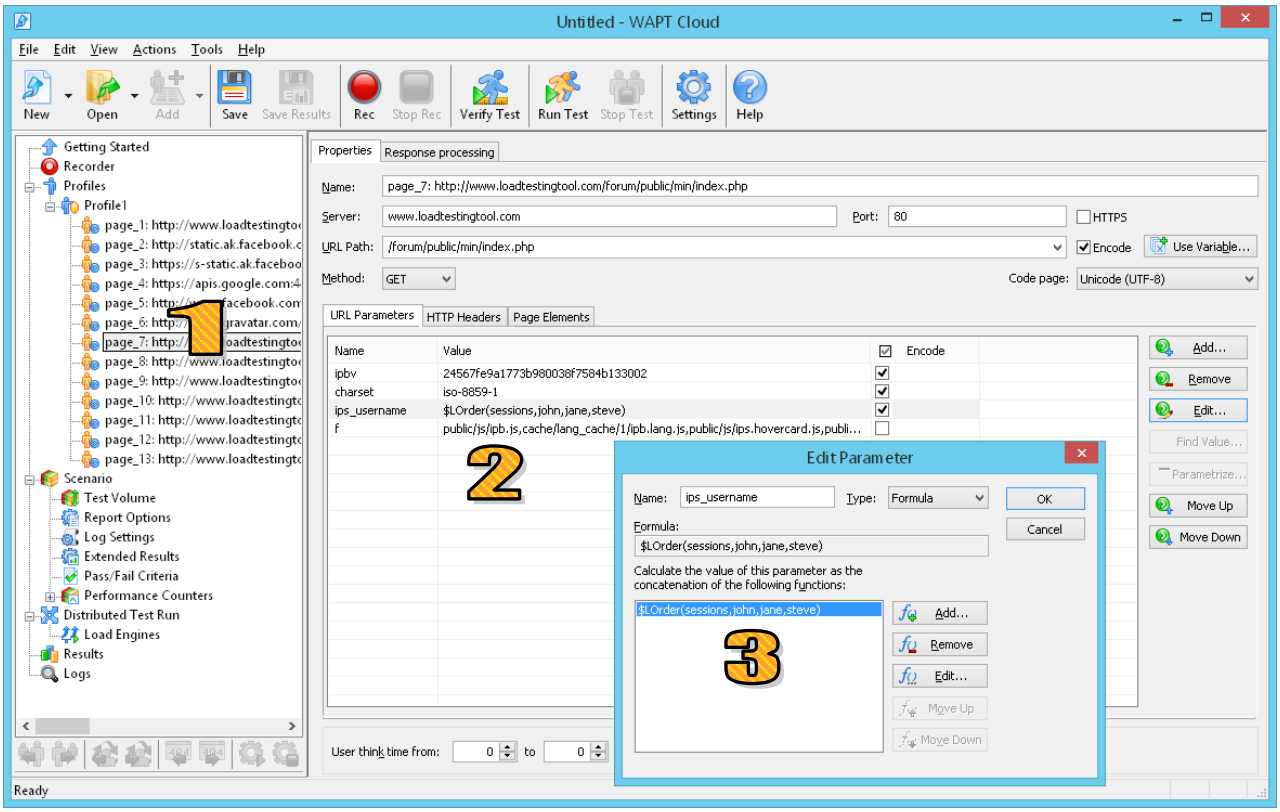
Note that WAPT Cloud works as a proxy between the browser and the target web site. Because of that you may experience some delays when working with your web application. Try to use it without haste. Do not click next link until you see that the current step has been fully recorded.

3 After you finish recording, click the **"Stop Rec"** button on the toolbar. WAPT Cloud will start processing the recorded content. It will separate images, .js and .css files from the page requests. The list of requests added to the profile will be much shorter than the original one. This is because all requests to page elements are placed on the **"Page Elements"** tabs for their parent pages.

You can record several more profiles in a similar way, or proceed with just one.

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 your web application to the server. For example, many applications send user name and password as parameters of the login request.

There are no standard names and meanings for all parameters. Each web application can use its own custom ones to pass its specific data. The number of parameters, their names and values may be different for each request.

Initially all parameters have static values that were used in the recorded session. 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 do this with the help of special WAPT Cloud **functions**. This is called ***"Parameterization"***. The understanding of this concept is very important for the successful use of any load testing tool, including WAPT Cloud.

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).

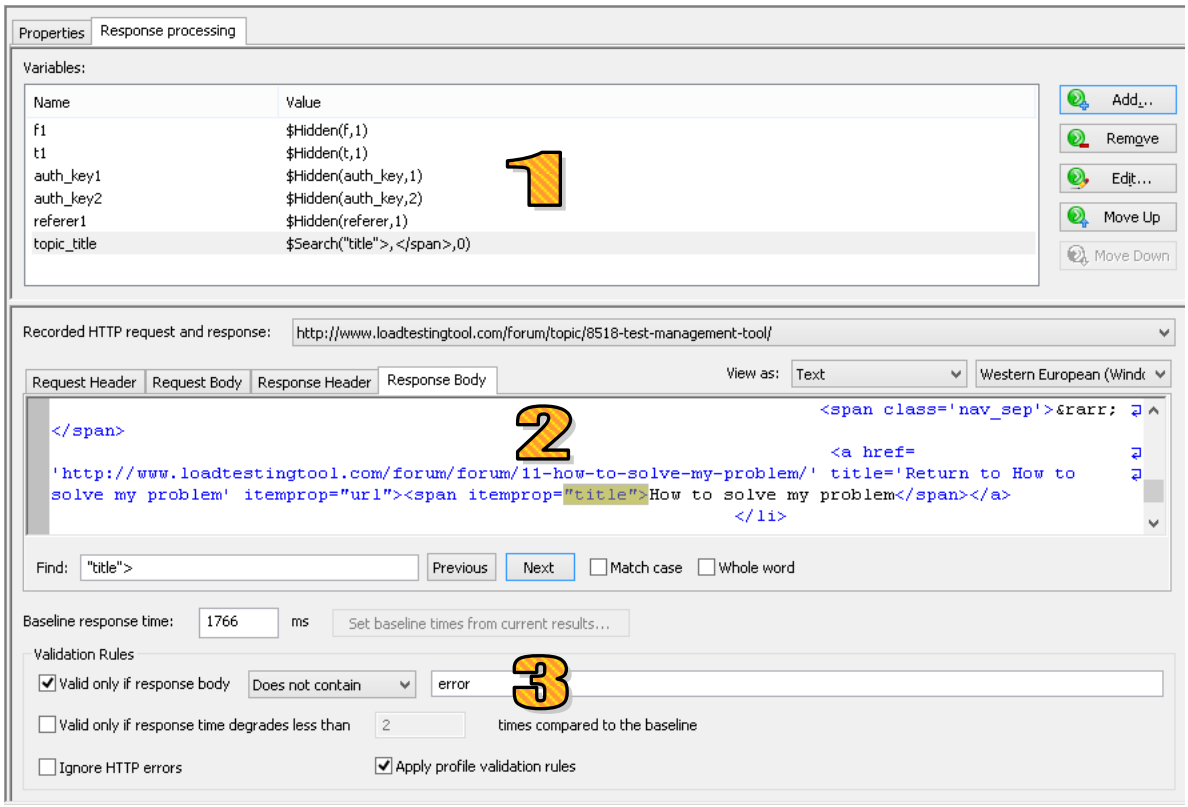
3 In the **"Edit Parameter"** dialog box you can specify how to calculate the value for the selected parameter. In the example shown on the screenshot above the value of the parameter is calculated with the help of the **"Ordered List"** function that takes one of three names from the list. You can specify 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 use **variables** assigned earlier.

The processing of server responses

Switch to the **"Response processing"** tab. Here you can specify how to handle response to the selected request inside each user session.

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

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 the 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 web site forum.

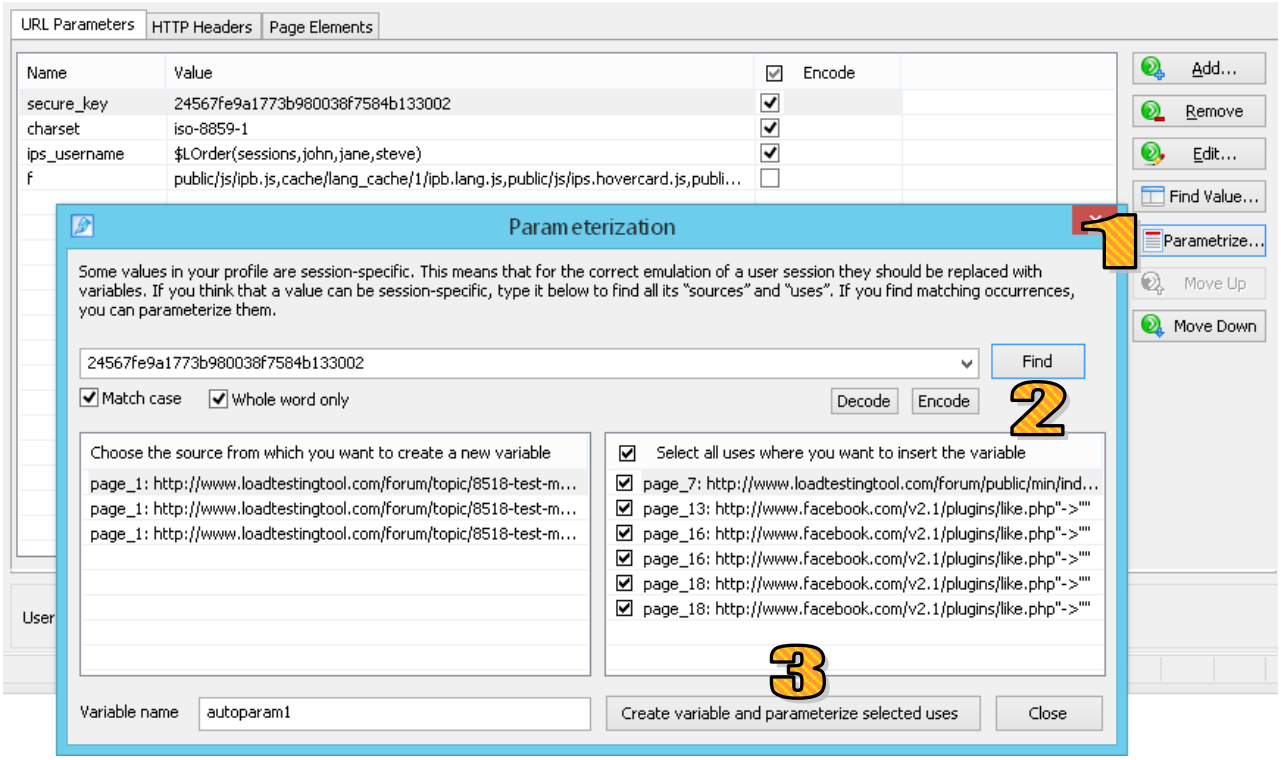
2 The tab contains full information on the initially recorded request and response to it. You can search for any text there. This is very useful, if you want to find the initially recorded value and its bounding text which can be used to specify the arguments of the **"$Search()"** function.

3 There is one more useful option on this page. You can specify custom validation rules here. You can make WAPT Cloud identify application-specific errors even if they are not reported through the HTTP response codes. In the above example the response is treated as valid, if it does not contain the *"error"* word inside.

Bunch parameterization

More than one request in a profile may require the same parameterization. A session-specific value may be produced by the server inside the page code and used as a parameter in several subsequent requests.

1 To speed up the work with such value, select it in the list of parameters and click the **"Parametrize"** button. The **"Parameterization"** dialog will appear. You can also open it through the **"Edit"** menu and enter the value manually.



2 Click the **"Find"** button to find all occurrences of the value in the profile. In the left pane of the dialog you will see all available "sources". This means all places in the server responses from which the value can be extracted. The right pane contains the list of all "uses" of the value.

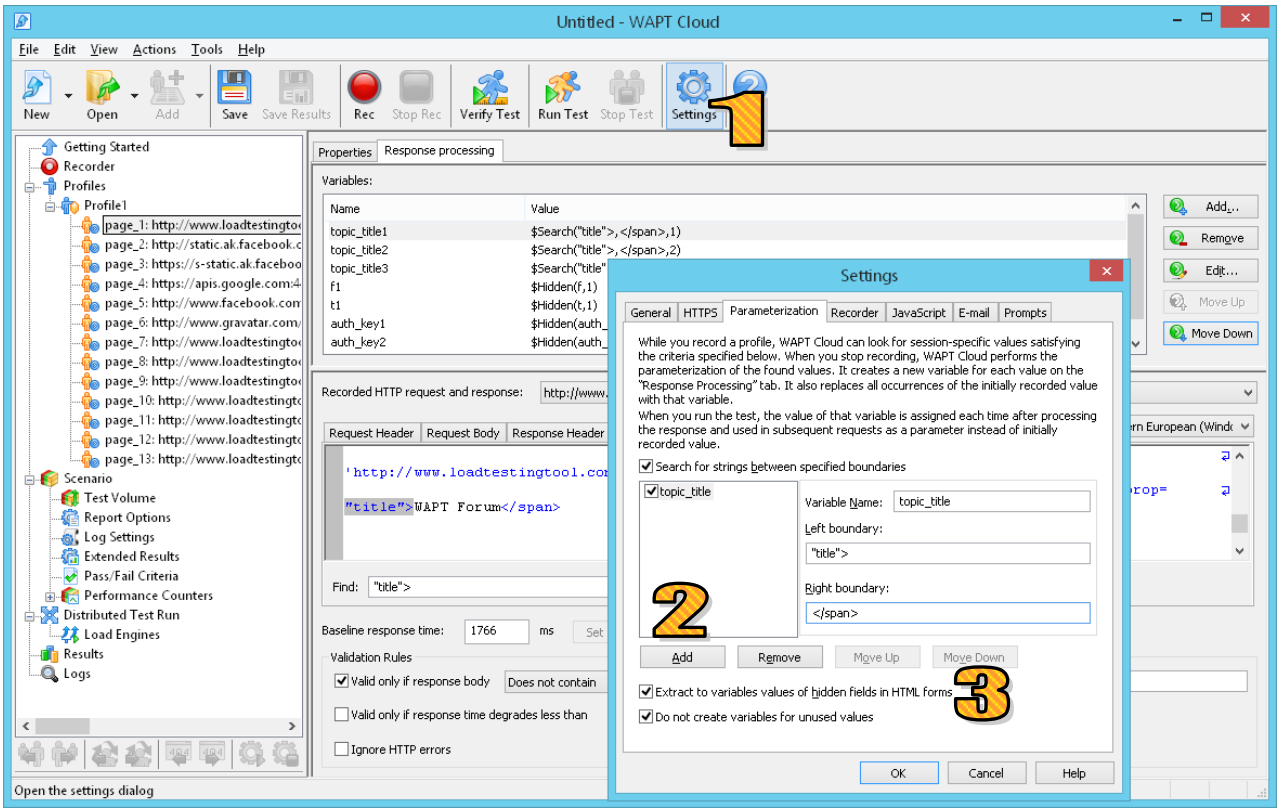
You should select one source and put checkmarks near all "uses" that you want to parameterize. You can click on items inside both panes to see the details in the main WAPT Cloud window. It is usually preferable to select the earliest source, i.e. the place where the value appeared for the first time.

3 Specify the variable name and click the **"Create variable and parameterize selected uses"** button. WAPT Cloud will automatically create a variable from the selected source and assign it with the help of the **"$Search()"** function. You will be able to see that variable in the processing of the corresponding response. It will also appear in all parameterized uses instead of the initial value.

Automatic parameterization

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

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



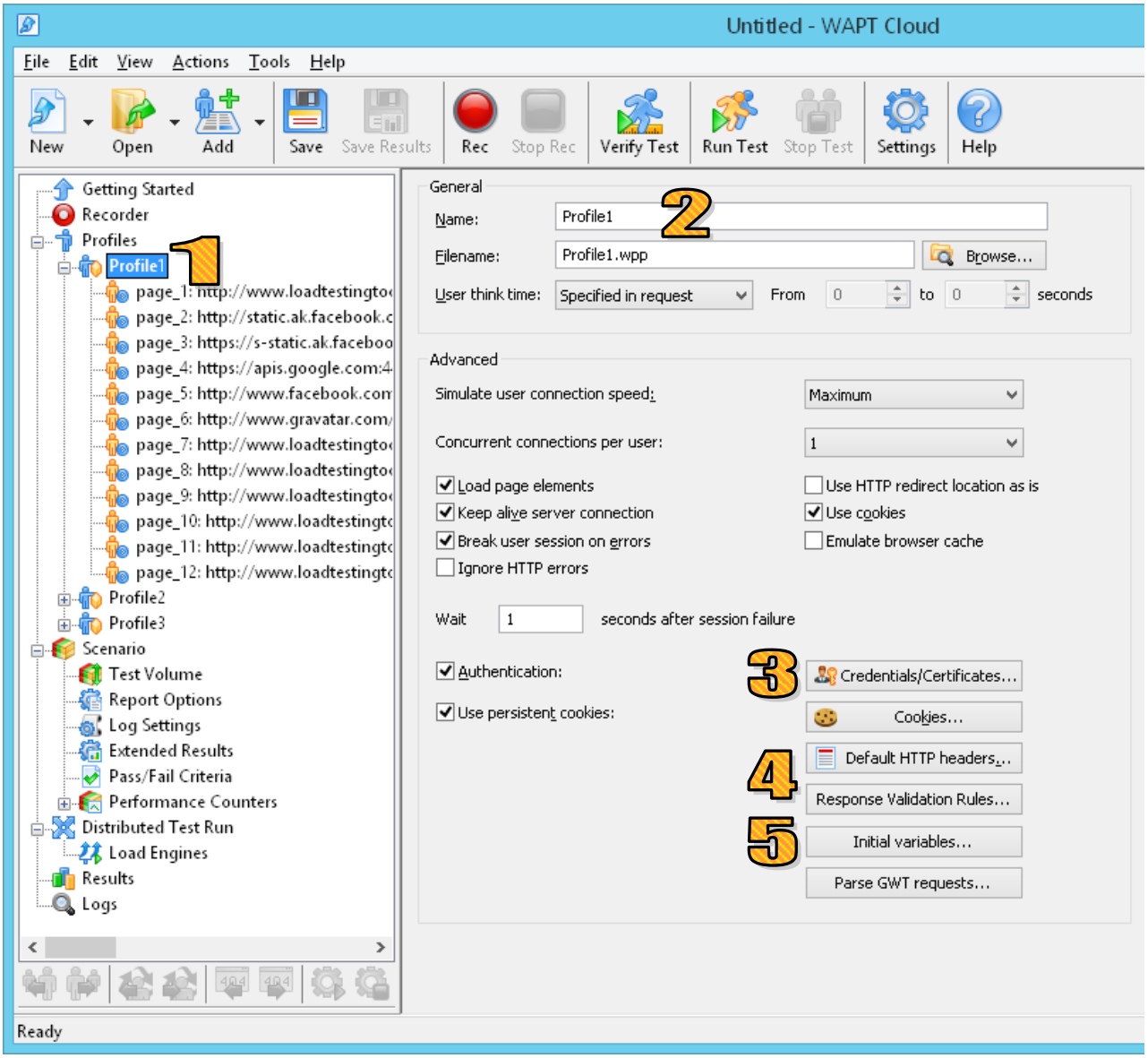
2 Click the **"Add"** button to create a new rule for extracting dynamic values contained on the recorded pages. For each value you should specify its left and right boundary. After you finish recording a profile, WAPT Cloud will check all the web site responses for such values and will create a new 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 Cloud 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 Cloud and assigned with the help of the **"Hidden"** function. You can also see how these variables are used in the parameters of the subsequent requests with the 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.

Not all hidden values are actually used in the test. To reduce the number of variables it is recommended to check the "Do not create variables for unused values" option. You can try temporary unchecking it to see how this affects the number of variables.

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.

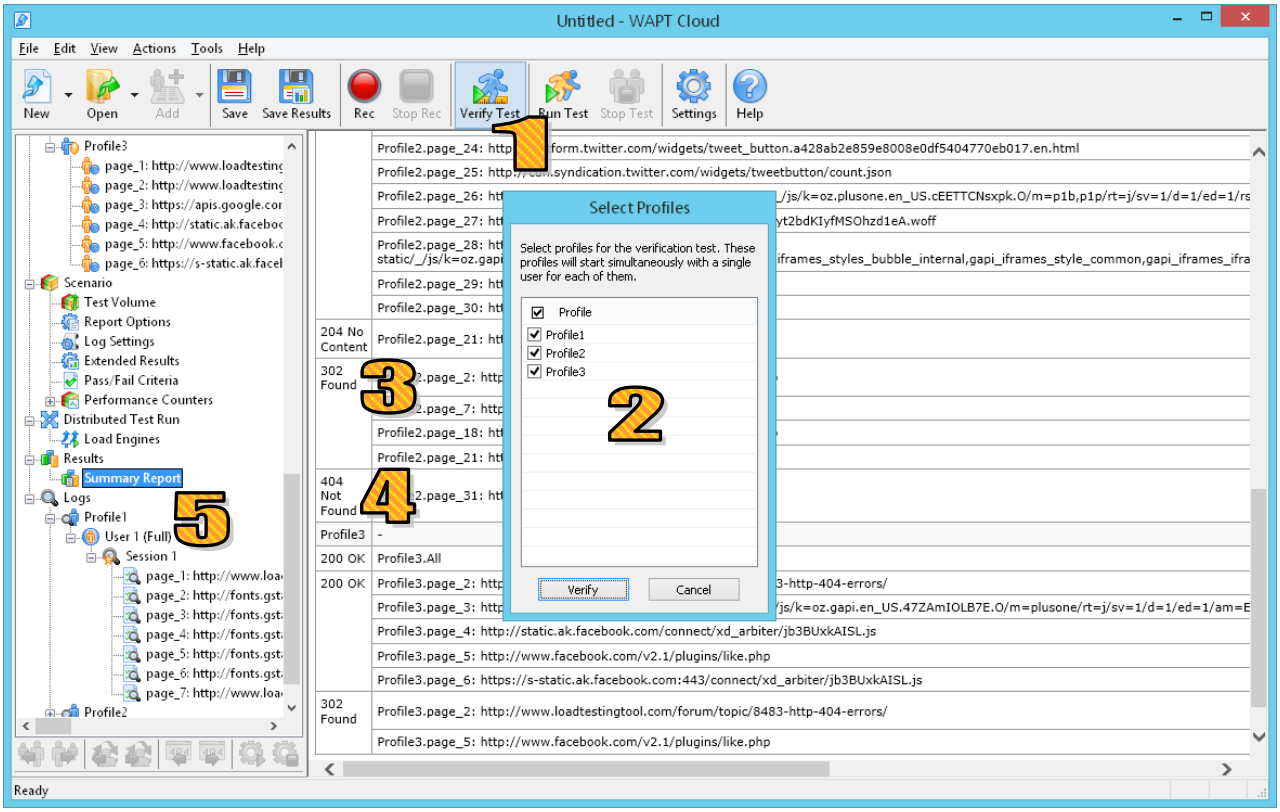
4 Note that **"Default HTTP headers"** and **"Response Validation Rules"** options can be overwritten in each request.

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 Cloud will let you select profiles for verification and will execute each of the selected profiles one time. When the verification is finished, WAPT Cloud 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.



3 Note that response codes starting with "3" (like 302) are not errors. These are HTTP redirects that are processed by WAPT Cloud 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 the 401 code, but it will be followed with the "200 OK" code. If you want to check this sequence step by step, you can use the logs.

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 the same problem there, your site contains a broken link.

If you see a "Network error", this probably means that WAPT Cloud 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 the 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 under 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.

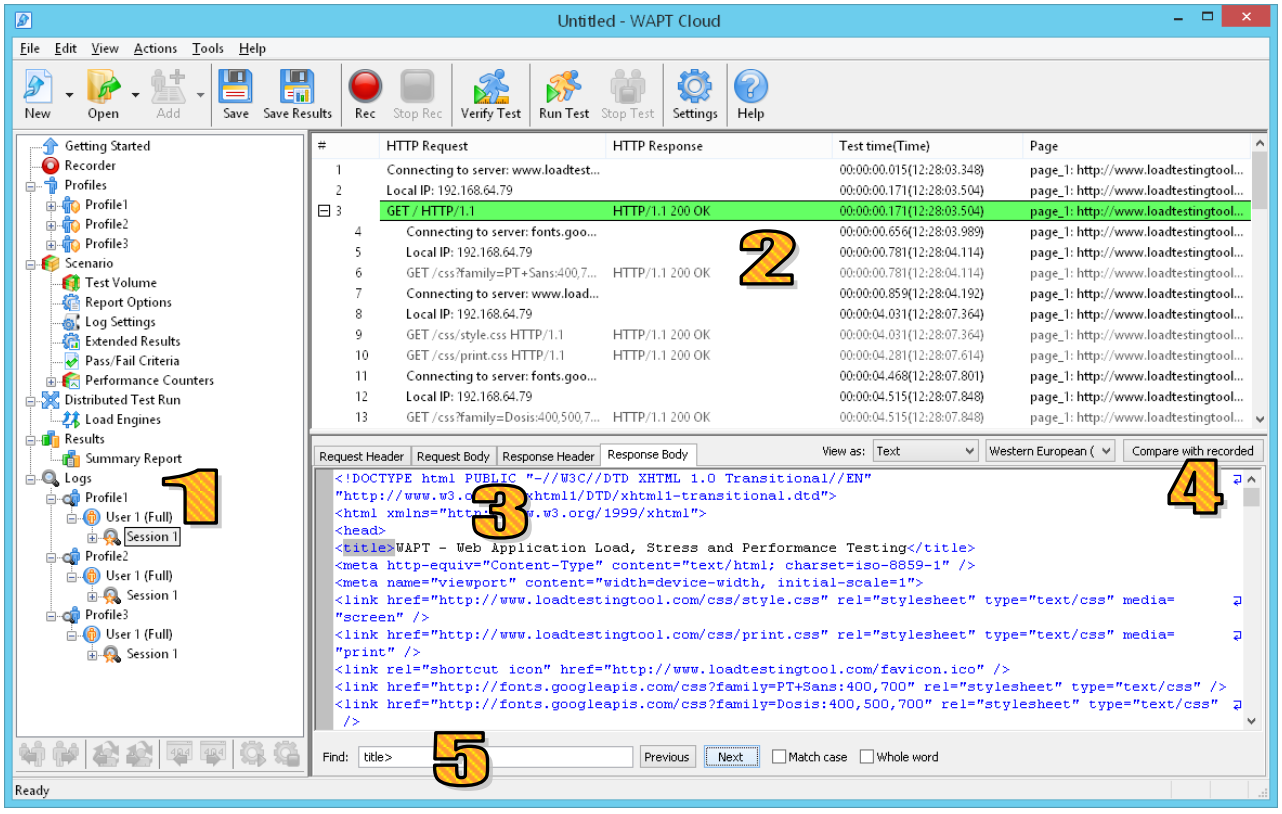
Log viewer

Log Viewer provides the detailed information on all requests, responses, and errors appeared during the test run or verification. This information is structured with the 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 the following 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 of the request or response with the initially recorded content. This way you can find the differences and identify session-specific values. You can also check if the server produced a significantly different content, which may indicate a problem.

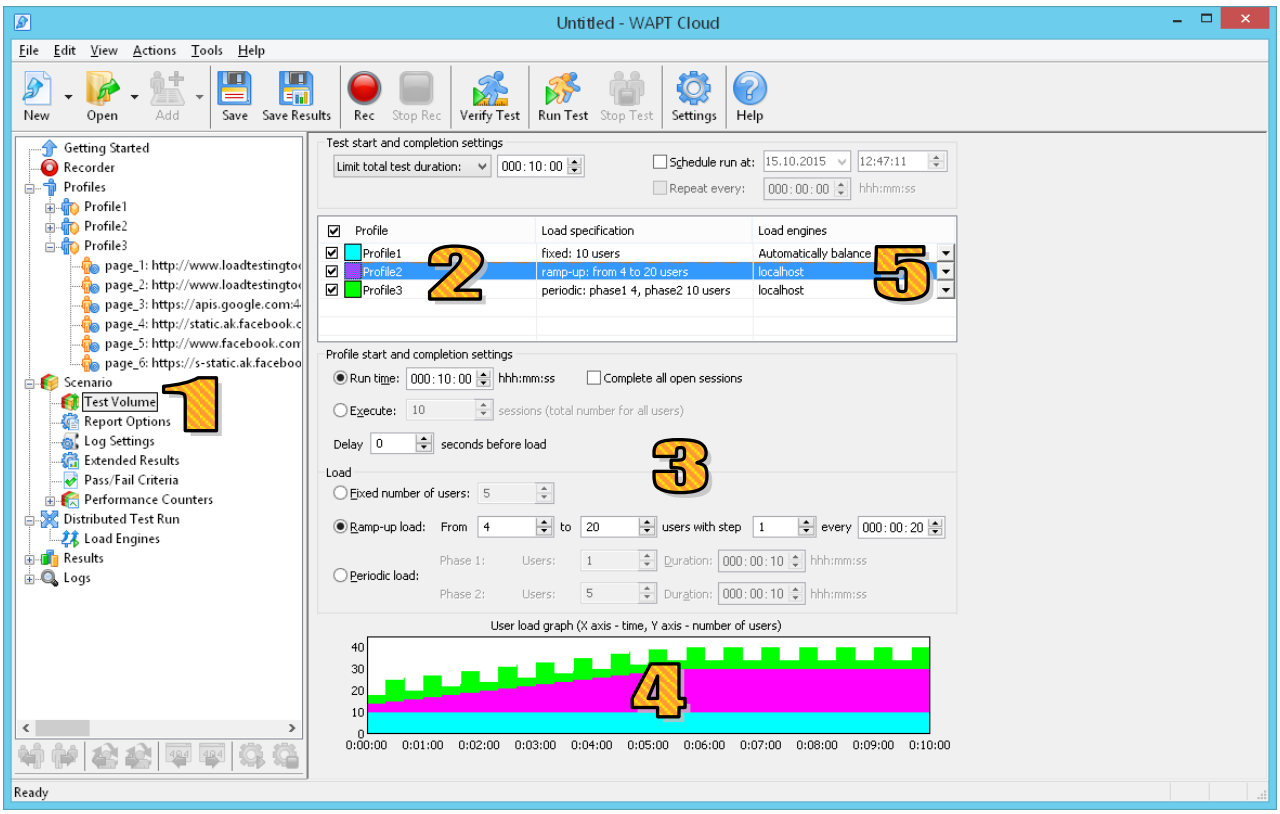
5 The useful search option is also available here.

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 Settings"** page. You can save all log files by choosing **"File | Save Logs…"** from the menu.

Test Volume

After making 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 edit 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 ramp-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 You can also specify different load engines for different profiles. By default they are distributed automatically, which means that all engines execute all profiles.

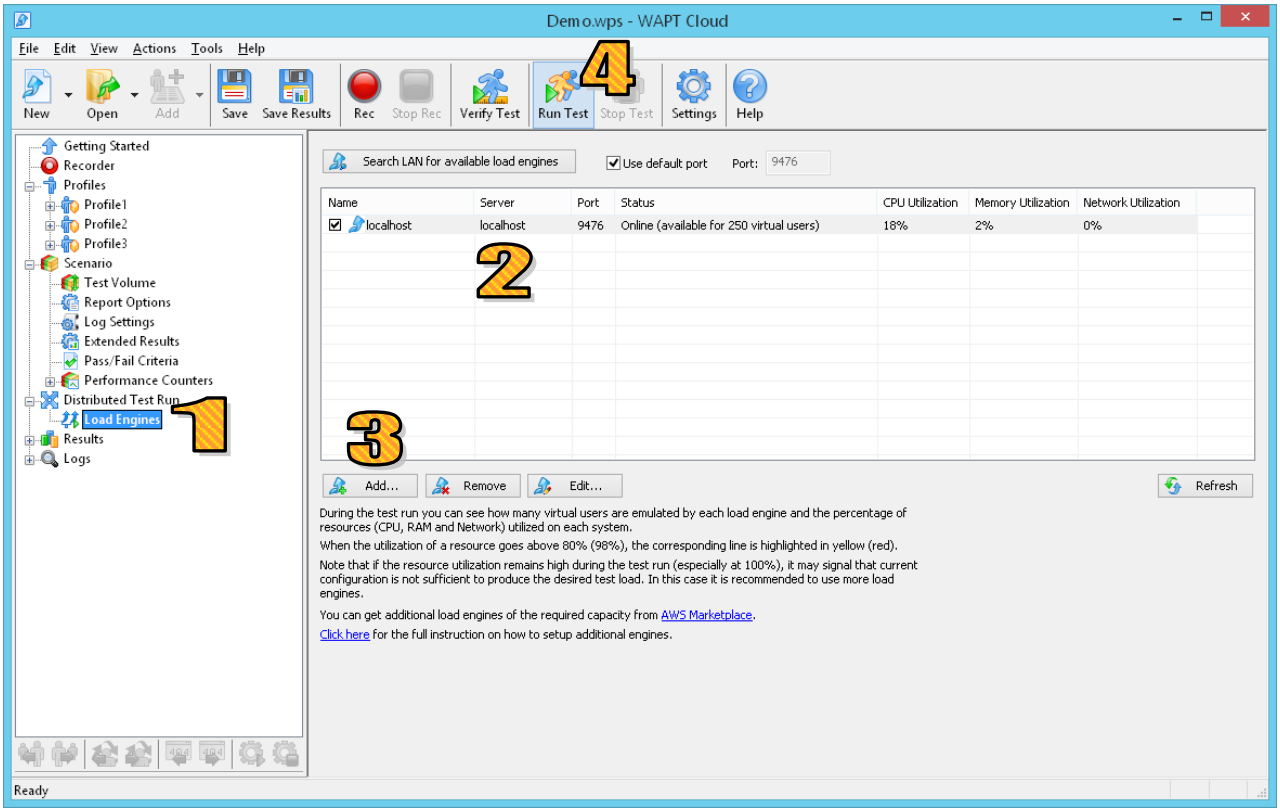
Now we have finished designing our test, so we can save it. Click the **"Save"** button on the toolbar to save your test scenario to a file. All profiles will be also saved to separate files in the same folder. Keep all these files, if you want to open the same test in the future.

Selecting load engines and starting the test

You can run tests with up to 250 virtual users with the help of the WAPT Cloud instance alone. The load will be generated from the same virtual system.

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

2 You will see the list of available engines and the capacity of each one in the list in the right view. Initially only the **"localhost"** engine is available.



3 If you want to run test with higher number of virtual users, start more engines available on the AWS Marketplace and click the **"Add…"** button to add them to the list. You will only need to specify the IP of each engine. You can take it from the EC2 Console where all your running instances are listed.

Cloud Engines are available at capacities of 2,000, 5,000 and 10,000 virtual users per system. The total generated load can reach up to 1 million users.

4 Put checkmarks near engines that you want to use in the test. You should check at least one. Click the **"Run Test"** button on the toolbar to start your test.

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 appear.

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 area. 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.

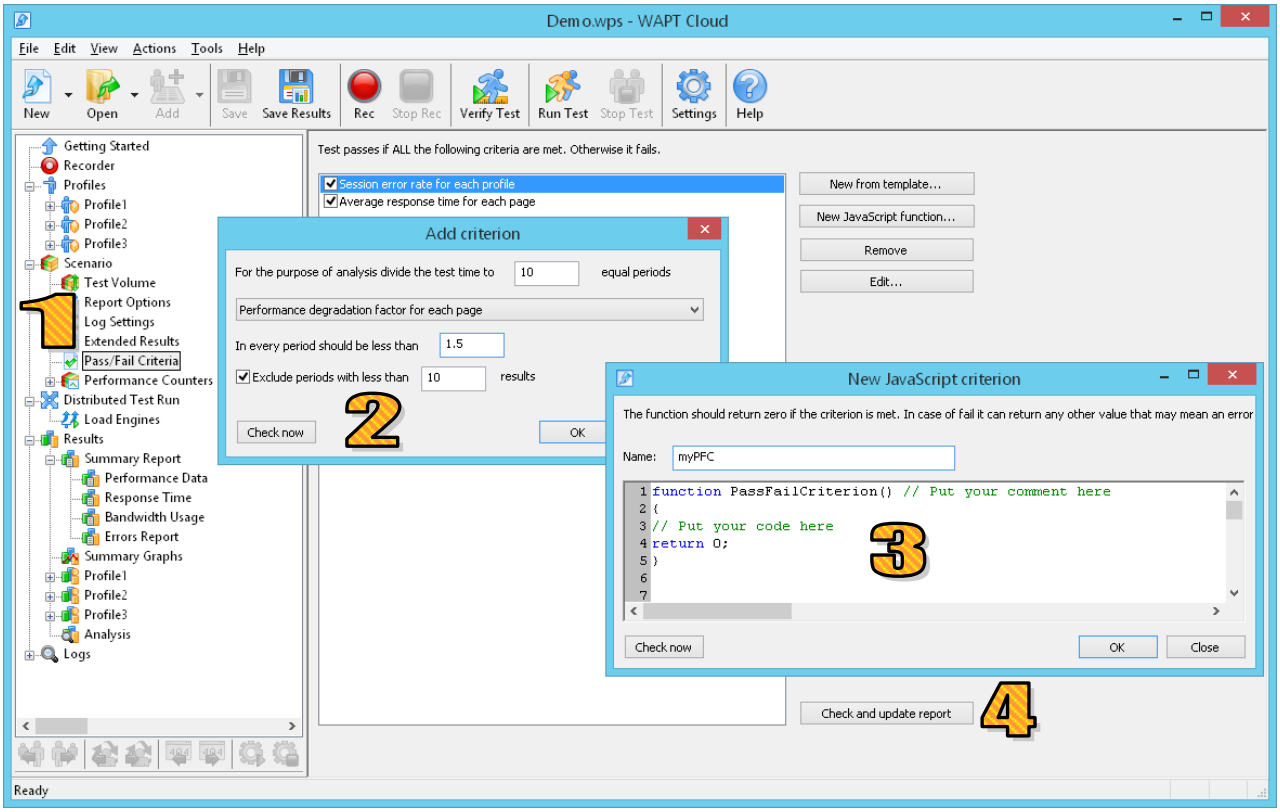
In the above example, the number of pages per second (orange graph) is specified on the left scale, whereas the number of active users (black graph) is specified on the right scale.

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 Cloud at any time again, browse graphs and work with the results like right after the test.

Pass/fail criteria

If you have exact requirements for the performance of your web application, you can make WAPT Cloud automatically check all the conditions to mark each test as passed or failed.

1 Select the **"Pass/Fail Criteria"** item in the left view. The right view will contain the list of criteria that will be applied to the test results.



2 Click the **"New from template…"** button to add a criterion basing on the standard templates. Each of them checks one of the parameters, such as the error rate or the response time.

3 You can also use criteria implemented with the help of a JavaScript code. This will let you perform very deep and specific analysis of the test results.

4 Criteria are applied to the test results automatically on the completion of each test. The test is treated successful only if all criteria are met. You can see the result in the Summary Report. You can also click the **"Check and update report"** button to apply the criteria to the latest test results. This is useful if you are fine-tuning your set of criteria.

This is what you will see in the report in case of the failure or success respectively.

Test result: FAILURE

Pass/Fail Criteria

Name	Result
Session error rate for each profile	SUCCESS
Average response time for each page	FAILURE

Test result: SUCCESS

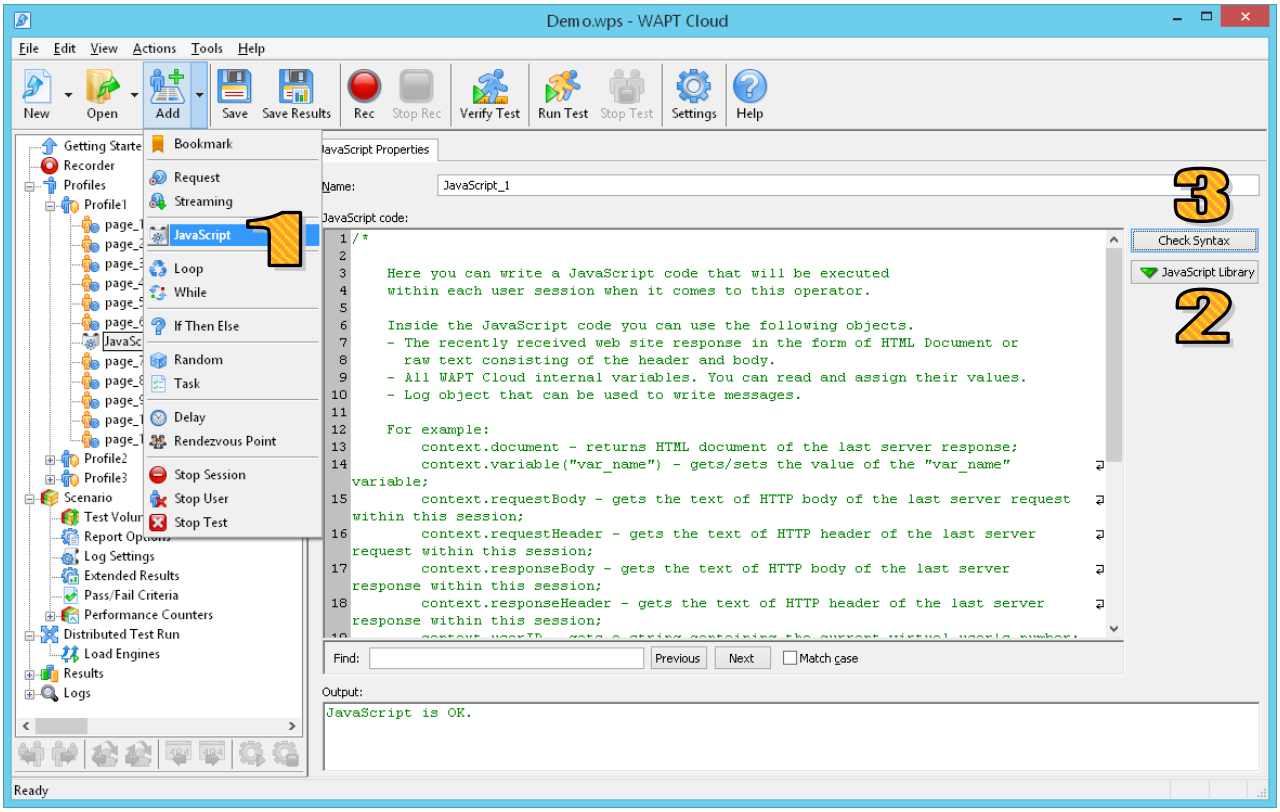
Pass/Fail Criteria

Name	Result
Session error rate for each profile	SUCCESS
Average response time for each page	SUCCESS

The use of JavaScript

You can use JavaScript code for the calculation of any values inside user sessions. This may be necessary if you need to emulate complex data processing on the client.

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 short instruction on how to use this feature.



2 In your code you can use functions defined in the WAPT Cloud JavaScript library. Click the **"JavaScript Library"** button to add files with more functions.

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

Note that JavaScript code can be used only for calculations. The results of such calculations should be assigned to WAPT Cloud variables. You can use those variables in subsequent requests. However you cannot initiate new requests or use GUI functions in the code.

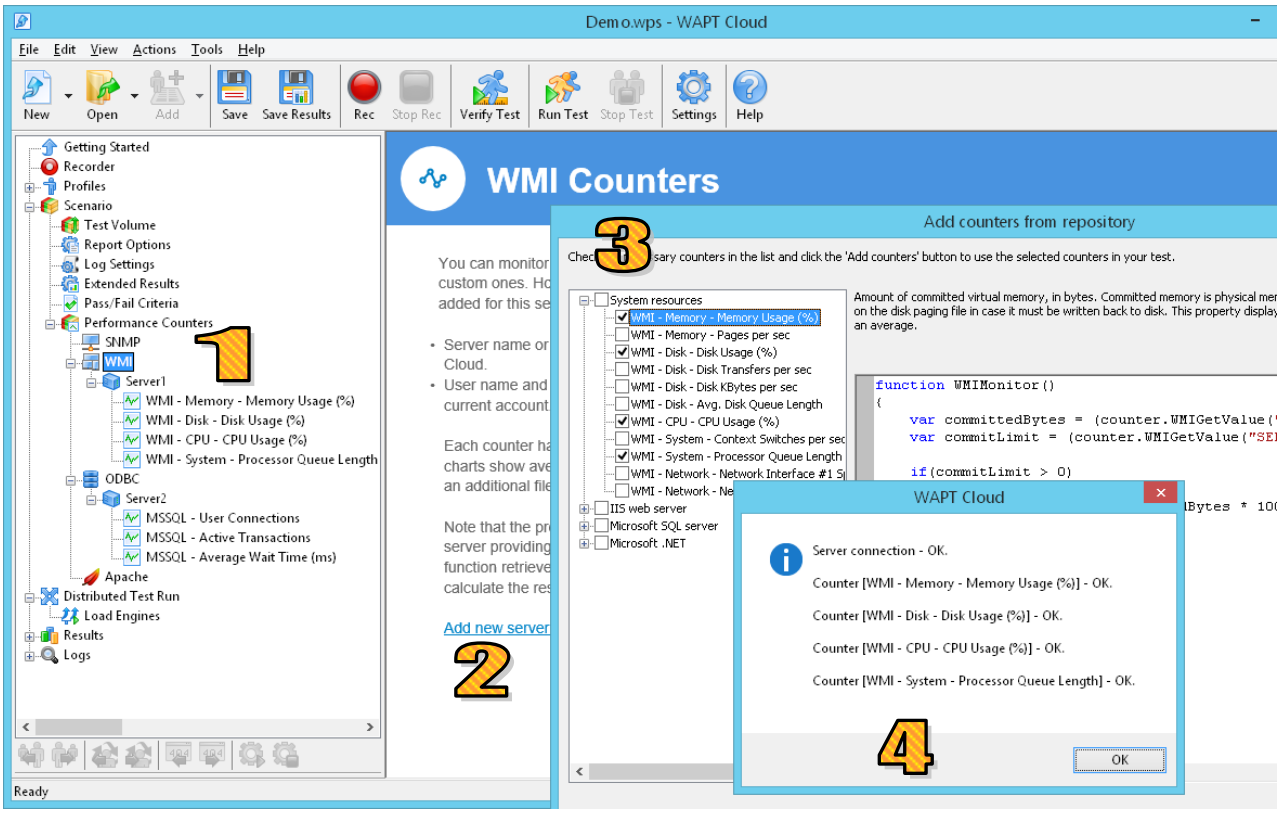
There is another way to use JavaScript in your profiles. You can call functions defined in the JavaScript library directly when you specify how to calculate values for the request parameters and variables. You can do this with the help of a special internal function called "JavaScript". It takes the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

Performance counters

In addition to the client-side metrics, such as response time, WAPT Cloud can collect performance information directly from the loaded servers. This information is added to the test report along with other parameters. You can also see it on graphs.

1 Expand the **"Performance Counters"** item in the left view. You will see the counters grouped by the interface type.

- **SNMP** is common for all types of UNIX systems; it can be enabled on Windows as well;
- **WMI** is native for Windows servers;
- **ODBC** is used to monitor database performance;
- **Apache** is specific for Apache web server.



2 Select an interface in the tree view and click the **"Add new server…"** link in the right view. This will create a new server. You can add counters for it.

3 WAPT Cloud has a set of predefined counters for a number of server tools. You can add and use them without any modification. Click the **"Add counters from repository…"** button on the server properties page for that.

Each counter is implemented as a JavaScript function that returns counter value. You can use the implementation of the predefined counters as examples and construct your own advanced counters in a similar way.

4 Click the **"Test server connection"** button in the server properties page to check that the counter retrieval works properly.