

# Load Test Tutorial

This tutorial demonstrates how to use Load Test to test service and Web interface functional tests. In this section:

- [Tutorial Overview](#)
- [Common Workflow For Test Scenarios that Include Web Functional Tests](#)
- [Common Workflow For All Other SOAtest Test Scenarios](#)
- [Creating a Web Application Functional Test](#)
- [Creating a Service Functional Test](#)
- [Creating and Performing a Load Test \(for Web and/or Service Functional Tests\)](#)
- [Customizing Load Test Profiles and Scenarios](#)
- [Viewing Reports](#)
- [Saving the Project as a .It File](#)

## Tutorial Overview

Once functional tests have been created in SOAtest, the next step is load testing with Parasoft Load Test. Load testing allows you to emulate conditions of heavy usage, which can expose bugs that may only surface under these conditions. You can load test Web, SOA, and combined end-to-end tests (test scenarios that extend beyond the message layer through Web services, JMS, web interface, database, etc.). [Watch the video tutorial](#).

In addition, Parasoft Load Test includes a framework for load testing any component that implements the Parasoft load test component API; for example, it can allow performance and concurrency testing of JUnits or load testing with lightweight Socket-based components that implement the Parasoft component API. This allows the load test to be specialized and tailored for the various unique complexities that organizations face in performing performance validation.

If you have load tests configured in previous versions of SOAtest and WebKing, they can be imported and are fully supported.

Parasoft Load Test allows you to have full control over all aspects of the load testing process including the following areas:

- **Enslaving Multiple Machines (Clustering):** You can enslave multiple machines (running Load Test) on your network to generate larger amounts of load than what a single machine can generate. Click on the **Machines** folder in the load test window and explore the GUI that appears. For each machine, you have the option for High Throughput mode, which generates higher load intensities using the same hardware by disabling certain response processing operations. See [Running Load Tests on Remote Machines](#) for details.
- **User Profiles:** Creating user profiles allows you to directly relate your load test back to your functional tests. This means that once you have created your functional tests, no further work is required to begin running it under load. Double-click the **Profiles** folder in the load test window and view each of the profiles that have been created.
- **Custom Scenarios:** Load Test provides four default load testing scenarios (Bell, Buffer Test, Linear Increase, and Steady Load) or allows you to create your own custom scenario. These scenarios can be created to emulate possible real life scenarios that may occur during normal usage. Click on the **Scenarios** folder and view the scenarios that are provided for this example.
- **Monitors:** Monitors can be added to Load Test to monitor various system resources as your load test occurs. Right-click on the **Monitors** folder to view the monitors that are available to be added. Load Test supports SNMP, Windows Perfmon, and JMX monitors. See [Using Monitors](#) for details.

## Video Tutorial: Configuring and Running Load Tests Using Existing SOAtest Scenarios

Common Workflow For Test Scenarios that Include Web Functional Tests

The typical workflow for load testing any SOAtest functional test suite that includes Web functional tests is:

- Record a web scenario using a browser.
- Customize the test scenario with the desired validations and extractions.
- Switch to the Load Test perspective in SOAtest.
- Select the appropriate test suite in the Load Test Explorer and click the **Validate for Load Testing** toolbar button.

If the execution succeeds, it is likely that no further scenario configuration/adjustment is needed and your scenario is ready for load testing by Parasoft Load Test. If the execution fails, then you can use SOAtest to configure your test suite so that it is ready for load testing. For instance, configuration might be required for variable URL parameter values that are passed across user actions within a certain browser scenario. For more details, see the Load Testing section of the SOAtest User's Guide.

Start Parasoft Load Test and create a new load test scenario for the .tst file that you created in SOAtest.

This same workflow applies to end-to-end test SOAtest test scenarios (test scenarios that extend beyond the message layer through Web services, JMS, web interface, database, etc.). Parasoft Load Test will drive concurrency out of any tests you may have in your SOAtest functional test suites.

Common Workflow For All Other SOAtest Test Scenarios

The typical workflow for load testing any SOAtest functional test suite that DOES NOT include Web functional tests is as follows:

- Define and customize the scenario in SOAtest.
- Start Parasoft Load Test and create a new load test scenario for the .tst file that you created in SOAtest.

Creating a Web Application Functional Test

In this load testing tutorial we are going to use a Parasoft-owned site named **Parabank**, which is a mock banking web application. Before you begin this lesson, we recommend that you follow the **Web Functional Testing** section in the SOAtest tutorial to learn how to record browser test suites for use in Load Test.

Recording a New Web Scenario

To record a Web functional test that we will use for load testing:

- In SOAtest, choose **File > New > Test (.tst) file**.
- Enter `ParaBank` as the **File name**, then click **Next**.
- Select **Web**; Record web scenario, then click **Next**.
- Select **Record new web scenario**, then click **Next**.
- Complete the Record from a starting location wizard page as follows:
  - Enter `ParaBank Functional Test` in the **Test Suite Name** field.
  - Enter `http://parabank1.parasoft.com/` in the **Start Recording From** field.



only be able to generate the Steady Load scenario. In the Other Options panel, enable Start Load Test immediately and Record individual hits, then click Finish.  The 'Other Options' panel in the Confluence GUI has two checkboxes: 'Start Load Test immediately' and 'Record individual hits', both of which are checked. Below these are two tabs: 'Load Test' and 'Graphs'. The 'Load Test' tab is active and shows a progress bar and a 'Finish' button. The 'Graphs' tab is inactive.

When you click Finish, the Load Test will begin the specified load test. A new Load Tests tab displays in the left GUI panel and the Graph tab displays in the Load Test progress panel.  The 'Load Tests' tab in the left GUI panel shows a progress bar and a 'Graph' button. The 'Graph' tab in the Load Test progress panel shows a graph of the load test progress.

You can change the number and layout of graphs shown by right-clicking in the Graphs tab, then choosing one of the available layout options.  The 'Graphs' tab in the Load Test progress panel shows a graph of the load test progress. A right-click context menu is open over the graph, showing options for 'Layout' and 'Zoom'. The 'Layout' option is selected, and a sub-menu is open showing options for '1', '2', '3', and '4' graphs.

The load test can be stopped at any time, but we will let it run the full 2 minutes while monitoring and manipulating the in-progress details of the test. Looking at the Load Test Progress tab, note that the Graph tab shows the following: The Virtual Users curve climbs in a steady, linear fashion, in accordance to the Linear Increase scenario chosen in Step 7 of this lesson. If the Tests Completed and the Tests Started curves closely match each other, this indicates that the tests are being served quickly (i.e. the responses are received quickly). If there is a wider gap between these curves, the execution time of the tests is longer. Also note that the Snapshot tab displays the current active virtual users and the operations they are invoking. During test execution, the information in this tab is updated every three seconds. After the load test is complete, a Test Information summary is displayed in the Results panel which includes the name of the project, when the load test was started and finished, the scenario you chose, as well as any machines and profiles.  The 'Results' panel in the Confluence GUI shows a test summary. It includes the name of the project, when the load test was started and finished, the scenario you chose, as well as any machines and profiles. The summary is displayed in a table format.

You can also choose to view different statistical reports of the load test as well. To view detailed statistics of the load test, complete the following: Select Statistics from the Views menu in the Results panel.  The 'Views' menu in the Results panel is open, and 'Statistics' is selected. The 'Statistics' report is displayed in the Results panel, showing a table of test results.

When viewing the Statistics report, an Output Types menu is available. The Output Types menu determines what type of output report is displayed. Two types of reports are available from this menu, each of which displays different columns of information. The differences between report types are: 

- Generic Reports: Contains Test Suite name, Test Index, Test Name, Min/Max/Avg Time (ms), Run Count, and Failure Count.
- Network Client Report: Contains all of the information in Generic Reports plus Min/Max/Avg Ping (ms), Min/Max/Avg Request Size (bytes), Min/Max/Avg Response Size (bytes) Min/Max/Avg Total Size (bytes).

To access details about a specific test failure, double-click the related report row. You can determine whether a given test failed by locating the row that represents that test, then checking the number in that row's Failure Count column.  The 'Network Client Report' in the Results panel shows a table of test results. One row is highlighted in red, indicating a test failure. The 'Failure Count' column for that row shows a value of 1.

You can customize how a particular load test is run by customizing the profiles and scenarios you plan to use. You can determine the length of time a load test lasts, the distribution of virtual users, the hit rate over time and machines, and the distribution of user profiles over time. Watch the video tutorial.  The 'Load Tests' tab in the Confluence GUI shows a video tutorial link. The link is titled 'Watch the video tutorial' and points to a video that explains how to customize load tests.

Double-click the Profiles folder in the Load Tests tab and select one of the available test suite nodes. The configuration panel displays on the right. At the bottom of the panel, change the delay Value to 3 seconds. This may simulate how a user hesitates before making a decision about ordering a book. Select Linear Increase beneath the Scenarios node. The Linear Increase scenario controls display in the Results panel and the User graph displays the localhost curve. Drag and drop the endpoint of the localhost curve to the coordinates of 10 users at 2 minutes. From the Vertical scale drop-down menu, select 20. Click the More Points button. A point will appear at the center of the localhost line. Click and drag the new point to the coordinates of 10 users at 1 minute.  The 'Load Tests' configuration panel in the Confluence GUI shows the 'Linear Increase' scenario selected. The 'Value' field is set to 3 seconds. The 'Scenarios' node is expanded, and the 'Linear Increase' scenario is selected. The 'Linear Increase' scenario controls are displayed in the Results panel and the User graph displays the localhost curve.

Expand the Linear Increase node and select the QoS node beneath it. Summary and Details tabs display. Select the Details tab and click the New button. The Add Metric wizard displays.  The 'Add Metric' wizard in the Confluence GUI shows the 'QoS' node selected. The 'Add Metric' wizard is displayed in a modal window, and the 'QoS' node is selected in the 'Add Metric' wizard.

The Add Metric wizard displays.  The 'Add Metric' wizard in the Confluence GUI shows the 'QoS' node selected. The 'Add Metric' wizard is displayed in a modal window, and the 'QoS' node is selected in the 'Add Metric' wizard.

Select the Details tab and click the Finish button. In the metric configuration panel that opens, enter Execution Time Requirement in the Name field. For the Statistic drop-down menus in the right GUI, select Avg. Exe. Time (ms) and the less than symbol (<), and then enter 200 in the text field.  The 'Add Metric' wizard in the Confluence GUI shows the 'QoS' node selected. The 'Add Metric' wizard is displayed in a modal window, and the 'QoS' node is selected in the 'Add Metric' wizard.

This will cause

the load test result to `fail`; if the execution time is measured above 200 milliseconds. For more information on configuring QoS metrics, see [Customizing QoS Metrics for Scenarios](/display/SOAVIRT9103/Customizing+Load+Test+Parameters#CustomizingLoadTestParameters-Qos).

Select the **Linear Increase** node beneath the **Scenarios** branch and click the **Load Test** toolbar button. Load Test will begin the customized load test and the **Graph** tab displays in the right GUI panel.

Wait (2 minutes) for the load test to complete. While the load test is running, you can view various parameters within the **Graph** tab by selecting the appropriate checkboxes.

After the load test is complete, a Test Information summary is displayed in the right GUI panel which includes the name of the project, when the load test was started and finished, the scenario you chose, as well as any machines and profiles.

## Video Tutorial: Fine-tuning Load Test Configurations and Scenarios

<https://player.vimeo.com/video/123453408>

## Viewing Reports

Once the load test is completed, collected data must be analyzed in order to see how the application/service performed under load. Load Test gives you the ability to configure and generate load test reports.

In this section, you will learn how to view the Detailed Report and how to generate an HTML Report.

To view detailed reports, complete the following:

- Select **Detailed Report** from the **Views** drop-down menu in the Results panel. A **Graph** tab displays with various parameters.
- Within the graph you can perform the following:
  - To view different parameters to graph and view, select the desired parameter check-box located at the bottom of the Results panel.
  - To view multiple parameters in a logarithmic scale, select the **Log Scale** checkbox. A logarithmic scale allows you to see the shape of multiple curves on the same graph (even if the displayed values are very far from one another).
  - To view recorded details for entire graph, right-click any area of the graph, then choose **Show Recorded Details** from the shortcut menu. Error details are recorded by default. You can record success details as well; see [Customizing the Data Recording Parameters](/display/SOAVIRT9103/Customizing+Load+Test+Parameters#CustomizingLoadTestParameters-dataRecording) for details on how to configure this.

As you can see, you can gather various details of a load test from the **Graph** toolbar button.



Save this load test project by clicking the **Save Project** toolbar button.

Save the Project as a .It File

Save the Project as a .It File

In the dialog box that opens, enter a name for the project in the **File** name field and click the **Save** button. Load Test saves the project and adds a `.It` extension to the file name.