Load Test Command Line Interface - cli

This topic explains how you run a load test from the command line interface.

Starting the Load Test Server on Remote Machines

If you will be running tests on remote machines, Load Test Server must be running on those machines.

Windows

To start the Load Test Server on a Windows machine, enter a command in the format:

Windows: `lt -loadtestserver [portnumber]`

For example:
`lt -loadtestserver 10095`

> **Note**
>
> `portnumber` should be replaced by the port on the local machine to which the server is listening. For your command, enter whatever port number is appropriate for your test setup.

Linux/Mac

To start the Load Test Server on a Linux/Mac machine, enter a command in the format:

`loadtest -loadtestserver [portnumber]`

For example:
`loadtest -loadtestserver 10095`

> **Note**
>
> `portnumber` should be replaced by the port on the local machine to which the server is listening. For your command, enter whatever port number is appropriate for your test setup.

Working with Load Test Machines Located on Different Networks

A Load Test controller machine (the one that runs in GUI mode and from which you control the load test) can communicate with a Load Test server machine located on a different network *if the network equipment that bridges these networks allows communication on the port that the Load Test server machine is listening on.*

For example, if you run your Load Test controller on network A and your Load Test server is started on port 10095 by a “lt -loadtestserver 10095” command on network ,B you need to make sure that the communication on port 10095 between networks A and B is not restricted by your network equipment. Allow any TCP traffic exchange for that port. Talk to your system or network administrator about the network configuration issues.

Running Load Tests

To perform the test, you create a script that specifies what `.lt` file to open and how to run the test, then you invoke that script whenever you want to run the test.

Create the Script

To create the script, create a text file (we’ll refer to this as `your_command_file`) that contains a command in the following format:

```
open FilePath

loadtest [-version]
[-localSettings path_to_local_settings_file] [-env environment_name]
[-minutes number]
[-vus number | -hps number]
[-teamServer path_inside_team_server]
[-report path_to_report_file]
[-html html_directory]
ScenarioName
```

The loadtest command runs a test suite load test using the specified scenario. See the Load Test Command Reference table below for the list and description of the loadtest command options.

Using Variables in the Script

You can declare and use variables in the script. A variable can be declared in the following way:

```
var VariableName1 = VariableValue
var VariableName2 = "Variable Value with Spaces"
```

An already-declared variable can be assigned a new name in the following way:

```
var VariableName1 = NewVariableValue
```

Sample Script

Here is a sample script (comments in the script are preceded with a "#" sign):

```
#
# Using script commands
#

var report-home = C:\CONTINUOUS_LOAD_TEST\ENTERPRISE_LOAD_TEST
var minutes = 1

var scenario = "Steady Load"
var base = tests/loadtester/accuracytest/tests
var category = Accuracy

var test-name = 1-Profile-HPS
open ${base}/${test-name}.lt
loadtest -minutes ${minutes} -allReports ${report-home}/%d/${category}/${test-name} ${scenario}
```

Run the Script

To invoke that script while starting a Load Test Server:

- Invoke Load Test from the command line by entering a command in one of the following format
  - Windows: `lt.exe -cmd -run your_command_file`
  - Linux/Mac: `loadtest -cmd -run your_command_file`

Note that the `-cmd` argument starts Load Test in command line mode, while the `-run` argument executes the specified Load Test script. Both arguments are required to run a script from the command line.

Load Test Command Reference

The following commands allow you to start the Load Test Server and run load tests:

Load Test Startup Command

Use these commands in either the Windows Command Prompt or a Linux shell.

Command	Description
Windows lt -loadtestserver [server number] **Linux/Mac** loadtest -loadtestserver [portnumber]	Starts the Load Test Server on the specified port number. To start the Load Test Server on port 8999, you would specify the port number in the command line argument such as: `lt.exe -loadtestserver 8999`

Load Test Script Commands

Command	Option & *option arguments*	Description
var		Format: `var variable_name = variable_value` Declares a variable name and assigns it a value. Use ${variable_name} syntax to get the variable value.
open		Format: `open path_to_load_test_project_file` Opens the specified project in Load Test. After a project is loaded, use the loadtest command.
loadtest		Format: `loadtest [-version] [-localSettings path_to_local_settings_file] [-env environment_name] [-minutes number] [-vus number | -hps number] [-teamServer path_inside_team_server] [-report path_to_report_file] [-html html_directory] [-allReports reportsPath][-trimReportSeconds headSeconds tailSeconds] [trimReportPercentage headPercentage tailPercentage]ScenarioName` Runs a test suite load test using the specified scenario and options. See the rows below for details.
	-localSettings *path_to_local_settings_file*	Specifies the path to the local settings file, which contains configuration and preferences parameters (see Local Settings Files).
	-env *environment_name*	Sets the active environment. Environments are a part of the SOAtest project that are used by the Load Test.
	-minutes *number*	Specifies the test duration in minutes. If you specify a test duration in this command, it will override the test duration specified in the scenario.
	-vus *number*	Sets the number of virtual users for each machine in the load test configuration to the given Number. The execution mode is switched to "Weighted Profiles" mode and each machine in the load test configuration is configured to run a steady load with the specified number of virtual users.
	-hps *number*	Sets the number of hits per second for each machine in the load test configuration to the given Number. The execution mode is switched to "Weighted Profiles" mode and each machine in the load test configuration is configured to run a steady load with the specified number of hits per second.
	-teamServer *path_inside_team_server*	Sends the necessary reports to the Team Server under the specified path. Uses the Team Server that is speci-fied in the local settings file. If no local settings are specified, uses the Team Server specified in the Load Test preferences. See Load Test Continuum for more details and examples. `-tcm path_inside_team_server` is deprecated but supported.
	-report *path_to_report_file*	Specifies that you want to create a binary report with the given name
	-html *html_directory*	Specifies that you want to create an HTML report with the given name in the specified directory.
	-allReports *reportsPath*	Specifies that you want to create a binary, XML and HTML report under the given directory path.
	-trimReportSeconds *headSeconds tailSeconds*	Specifies the duration (in seconds) to be trimmed from the beginning and the end of the report*. The seconds values must be non-negative integers. If the sum of the headSeconds and tailSeconds values is greater than the duration of the report, the trim operation will not be applied.

	-trimReportPercentage *headPercentage tailPercentage*	Specifies the duration (in a percentage relative to the duration of the report) to be trimmed from the beginning and the end of the report*. The percentages must be non-negative integers. If the sum of the headPercentage and tailPercentage is equal to or greater than 100, the trim operation will not be applied.

*Load Test report data is collected and saved in approximately 3 second intervals. The trimReport…

commands will discard all report collection points that fall into the specified trim intervals. Because the times of the report collection points and the specified trim values may not match, the actual duration trimmed from the beginning and end of the report may (and in most cases will) exceed the duration specified in the command line. The excess value trimmed from the head and the tail of the report will not exceed the duration of the data collection interval. For more details on how report fragments are saved, see Understanding Report Interval Selection.

Local Settings Files

Local settings files can control various Load Test options. You can create different local settings files for different projects, then use the `-localSettings` script option to indicate which file should be used for the current command line test.

If a parameter is specified in this file, it will override the related parameter specified in the Load Test preferences. If a parameter is not specified in this file, Load Test will use the parameter specified in the preferences. If there is no relevant parameter in the Load Test preferences, a default value setting will be used.

If you want to use a default setting, remove or comment out the setting from the file. The file contains name/value pairs and uses the following format:

`nameString=valueString`

Each name/value pair must be on a separate line. Use the hash “#” symbol to comment the name/value entries.

Sample

A sample local settings file may contain:

```
# This is a commment
echo=false
team.server.host=localhost
team.server.port=18888
team.server.user=
team.server.password=
```

Available Settings

Setting	Description
`echo=true|false`	If “true” Load Test will echo the settings to the console. Default state is true. If this setting is used, it must be the first entry in the local settings file.
`console.log=enable`	Turns on console logging in command line mode. See Console Logging section below. Default is disabled.
`team.server.host=host_name`	Team Server host name. If this setting is not specified, the Load Test Preferences Team Server setting will be used.
`team.server.port=port_number`	Team Server port number. If this setting is not specified, the Load Test Preferences Team Server setting will be used.
`team.server.user=user`	Team Server user name. Leave this setting empty to specify empty user name. If this setting is not specified, the Load Test Preferences Team Server setting will be used. This Team Server parameter is optional; it is not required with most configurations.
`team.server.password=password`	Team Server password. Leave this setting empty to specify empty password. If this setting is not specified, the Load Test Preferences Team Server setting will be used. This Team Server parameter is optional; it is not required with most configurations.

Console Logging

Console logging provides load test status information in command line mode as the tests are running. When console logging is enabled (see the `console.log=enable` local setting), the application prints the main load test run time parameters in the console. Load test status is printed in one line every 30 seconds. It contains the following information:

- System Time
- Percent complete
- Run time seconds completed/total
- Total tests passed/failed
- Average Virtual User count since last log
- Average Hit Per Second rate since last log

Example:

```
2011.04.21 17:54:56 MDT | 53% complete | 32/60 seconds | passed 245/failed 0 | avg. VUs 4 |avg. HPS 8
```