



LoadComplete
by **SMARTBEAR**

Getting Started

Copyright Notice

LoadComplete, as described in this online help system, is licensed under the software license agreement distributed with the product. The software may be used or copied only in accordance with the terms of its license.

© 2019 SmartBear Software. All rights reserved.

No part of this help can be reproduced, stored in any retrieval system, copied or modified, transmitted in any form or by any means electronic or mechanical, including photocopying and recording for purposes others than personal purchaser's use.

All SmartBear product names are trademarks or registered trademarks of SmartBear Software. All other trademarks, service marks and trade names mentioned in this Help system or elsewhere in the LoadComplete software package are the property of their respective owners.

LoadComplete uses the 7z.dll module that is part of the 7-Zip program. This DLL is licensed under the GNU LGPL. Its source code can be found at <http://www.7-zip.org>.

Table of Contents

LOADCOMPLETE OVERVIEW.....	4
Load Testing - Basic Concepts	4
Emulating Real-Life Conditions	6
LoadComplete User Interface	7
CREATING YOUR FIRST TEST	9
Preparing For Creating Your First Test	10
1. Creating a Test Project	11
2. Recording a User Scenario.....	13
3. Viewing the Recorded Scenario.....	20
4. Verifying Your Recorded Scenario.....	23
5. Creating a Load Test.....	25
6. Running Your Load Test.....	29
7. Analyzing Results	32
Report Panel.....	32
Details Panel	32
Resolving Issues.....	32
Where to Go Next	32
INDEX.....	34

LoadComplete Overview

Load Testing - Basic Concepts

LoadComplete is a tool for creating website load tests. Load testing consists in simulating multiple **virtual users** that send requests to the tested server concurrently and evaluating the server performance under the massive load.

Virtual users are not real users. LoadComplete simulates recorded requests and the tested server “thinks” that it works with an actual user. Each virtual user works according to its task, or **scenario**. A scenario is a sequence of requests sent to the server and the server’s responses to them. In other words, the scenario emulates user actions on the tested web site.

Load testing of websites with LoadComplete includes the following steps:

1. Recording a scenario.

You create scenarios by recording interactions with your tested web application or server. During the recording, LoadComplete captures all HTTP, HTTPS, and WebSocket traffic between your computer and the target server. To record a scenario, you need to start the recording, perform all desired actions against the tested web server and then stop the recording. When you start the recording, LoadComplete creates a load test based on the scenario. After you finish the recording, LoadComplete automatically saves it.

2. Modifying the recorded traffic (optional).

After you record a scenario, you may want to modify the recorded traffic. For example, you may want to remove unnecessary requests, change information sent to the web application and so on. You can do this in LoadComplete’s **Scenario** editor.

3. Verifying the recorded scenario.

It is a good practice to verify that the recorded scenario works fine. To do this, run the **Verify Scenario** command for the scenario. LoadComplete will run a scenario for one virtual user and check whether the recorded traffic simulates correctly.

Another way to check if the scenario works correctly is to run the single-user test that LoadComplete creates automatically after recording a scenario. The difference from verifying the scenario is that you can check if the scenario works correctly with various settings available for load tests (*SLA, Random think time*, and so on).

4. Creating load tests that will simulate recorded traffic.

You create a load test on the basis of a verified scenario. You can record several scenarios that will simulate various user actions on the tested web site and create a load test that will run these scenarios simultaneously.

You can also use a single-user test that LoadComplete creates automatically for a recorded scenario.

5. Assigning recorded scenarios to desired virtual users.

After you create a load test, you assign the desired number of virtual users to it. The maximum number of concurrent virtual users that can be simulated on a single workstation is 2000. To simulate more virtual users, you can distribute them among several workstations.

If you simulate traffic on Amazon cloud computers, you can assign **any** number of virtual users to the cloud computers in your load tests. LoadComplete will launch the needed number of instances and distribute the virtual users among them automatically.

Note: For Amazon cloud computers, you can manually set the maximum number of virtual users that can be simulated on a single instance (the default value is 400). LoadComplete will use the value to calculate the number of instances to launch from the machine image.

6. Running load tests.

When you run a load test, LoadComplete runs the Remote Agent processes (*RAS.exe*) to simulate virtual users. This process sends all the recorded requests to the target web server and then compares the received responses with the recorded responses. During the test run, LoadComplete neither opens the browser, nor loads any web pages to it.

Note: One Remote Agent process can simulate a limited number of virtual users. If your load test simulates more virtual users, LoadComplete launches additional *RAS.exe* processes. After the testing is over, LoadComplete closes all extra *RAS* processes.

7. Analyzing load test results.

After the test run is over, LoadComplete provides you with detailed information on simulated requests and received responses. It automatically shows the panel that contains detailed information on the test run and assists you in dealing with errors and warnings that occurred while the test was being executed (if any).

Note: You can export test results and send them via email.

You can follow the *Creating Your First Test* tutorial to learn more about creating load tests with LoadComplete. After completing this tutorial you will be able to create simple tests with LoadComplete, run them and analyze test run results.

Emulating Real-Life Conditions

After recording a scenario, you create a load test that will simulate the recorded actions for several virtual users (emulate a load for the server). You can specify not only the number of virtual users that LoadComplete will simulate, but also additional settings that allow emulating “real-life” conditions for the created load test.

Specifying such conditions, you can create more powerful and reliable load tests.

The main features that assist you in simulating realistic conditions for a load test are as follows:

- **Increasing the number of users.** With LoadComplete you can simulate not just a predetermined number of virtual users, but also a growing number that increases every time a specified period expires. This feature is called **load profile**. You use it to set not only the time period, but also the starting number of users, the increment number and the total number of users for the load test. Creating such tests is very helpful when you need to check how many users your web server supports. For more information, see *Defining Workload Profile* in LoadComplete documentation.
- **Emulating a massive load.** LoadComplete lets you simulate a massive load for the tested web server using the special **continuous load** feature. It allows executing the specified test nonstop for a number of times unless the specified period expires. For example, if your test takes 2 minutes to be executed, but you want it to be executed in 10 minutes, you can easily do that using the *continuous load* feature. For more information, see *Setting The Overall Test Duration* in LoadComplete documentation.
- **Parameterizing tests.** While recording a scenario, you may need to specify some parameters that will be used for further test runs. For example, you can enter some search terms, user names and so on in the application's fields. However, it is not a good practice to play back a test with recorded data for each user as it does not simulate the real-life conditions. To solve the problem, you can parameterize your load test using special variables. A variable can store desired parameterizing data and your test's requests can use this variable during the test run. For more information, see *Variables - Overview* in LoadComplete documentation.
- **Emulating browsers, network connection bandwidth and browsing speed.** Real users visit a web site using various browsers, the bandwidth of their Internet connection varies and they spend variable time on each page. You can configure all these parameters when creating load tests in LoadComplete:
 - You can specify browsers and connection bandwidth for virtual users.
 - You can also specify browsing time for each tested page (we call it **page think time**, because it simulates the period when a user is viewing the page and thinking).
 - You can easily randomize the think time for simulated scenarios to emulate real user activity better.

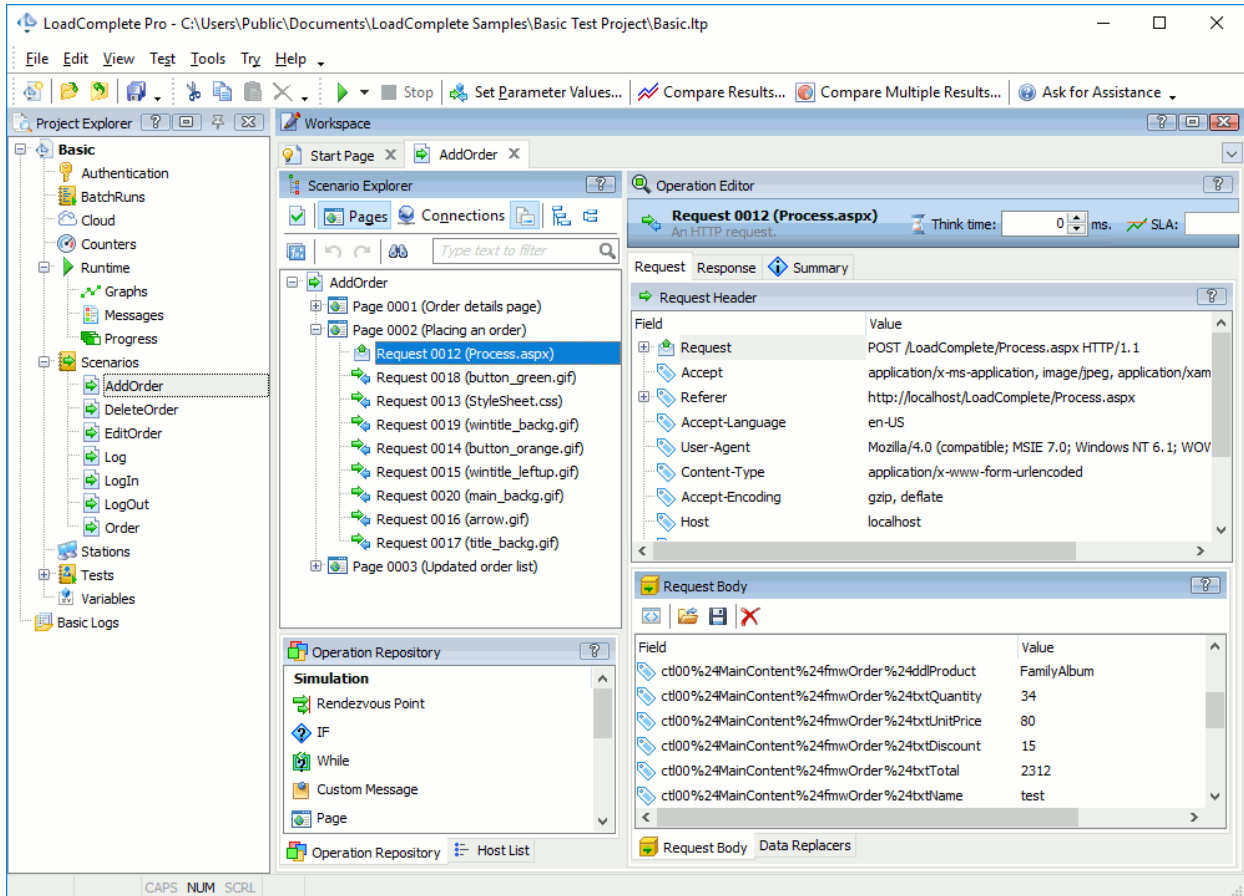
See *Creating Load Tests* in LoadComplete documentation.

- **Simulating concurrent requests.** To download a web page, modern web browsers send requests to the server using several simultaneous connections. These parallel requests download images, scripts, CSS files and other resources located on the page. Unlike many other load testing tools, LoadComplete automatically detects and simulates parallel requests, that is, it sends requests like a

browser would. This makes the simulation closer to real-world conditions. See *Simulating Parallel Requests* in LoadComplete documentation.

LoadComplete User Interface

The LoadComplete user interface is organized as a number of panels that assist you in creating and running load tests. Below you can see a sample image of the LoadComplete interface:



Each panel in LoadComplete serves a certain purpose. For example, the *Project Explorer* panel provides access to the test project currently opened in LoadComplete. The *Workspace* panel is a container for editors where you can create and modify tests and explore test results. To learn more about panels and their functions, see *LoadComplete Panels and Editors*.

The size and layout of panels are not prescribed. You can change the size of panels by dragging the separator between them, changing their position and docking. Panel docking in LoadComplete is similar to docking allowed in Microsoft Visual Studio. For more information on customizing panel layout, see *Arranging Columns, Lines and Panels*.

LoadComplete menus and toolbars provide you with commands that allow you to perform various actions over your tests. For more information on menus and toolbars, see *LoadComplete Menus* and *LoadComplete Toolbars* help topics accordingly.

You can change location of toolbars, move items from one menu or toolbar to another, hide items and so on. For more information on working with toolbars and menus, see *Customizing Toolbars and Menus*.

Creating Your First Test

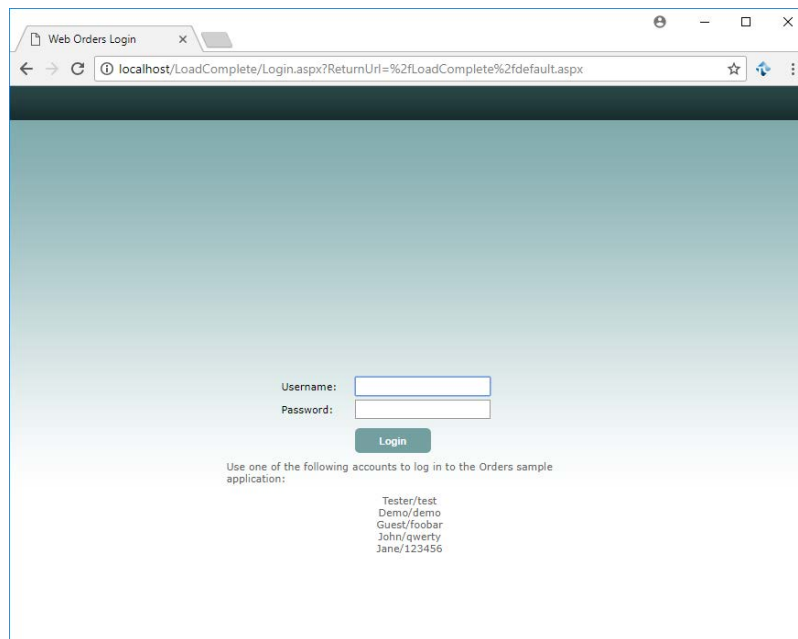
This section provides a step-by-step tutorial that demonstrates how to record a scenario in LoadComplete, create a load test that simulates the recorded scenario, run the load test and analyze its results. The test will simulate actions of several virtual users over a web site and check the target server responses.

Sample Application

In this load testing tutorial we will demonstrate all of the actions described above using the sample **Web Orders** application. This application is supplied along with LoadComplete and installed automatically during the product installation. This sample requires that your local machine have the following software installed:

- Microsoft Internet Information Services (IIS) 7.0 or later.
You may also need to enable certain IIS features. See the *LoadComplete Basic Test Sample* section in the *LoadComplete Documentation*.
- .NET Framework 3.5 or later.
- Microsoft Data Access (if you need to work with the database files).

Before you start creating your first test, make sure our sample application is installed and functions properly. To do this, enter <http://localhost/LoadComplete/> in the address bar of your web browser and press Enter:




If the application is not available, make sure that all the needed requirements are met and then reinstall LoadComplete.

Preparing For Creating Your First Test

Before you start recording your first test, consider the following prerequisites:

- LoadComplete must be added to the list of trusted application of your firewall and antivirus (see *Settings Required to Use LoadComplete* in LoadComplete documentation).
- If you use a web browser other than Internet Explorer, Edge or Google Chrome, configure it to use the system proxy settings to connect to a network.

To configure Firefox:

1. In Firefox, click  **Open menu** and then click **Options**.
2. In the **Options** dialog, locate the **Network proxy** options. You can type *proxy* in the search box on the Options page to find the options faster.
3. Click **Settings**.
4. In the **Connect Settings** dialog, select **Use system proxy settings**.
5. If you use Firefox 65 or later, clear the **No proxy for** list.
6. Click **OK** to close the dialog and save the changes. Then close the **Options** dialog.

To configure a third-party web browser:

For information on how to configure your web browser to use the system proxy settings, see your web browser's documentation.

- If you use Firefox 49 or later, we recommend that you configure your Firefox browser to use a Windows certificate store:
 1. In Firefox, type `about:config` in the **Address** bar. This will open the Firefox preferences page.
 2. Locate the `security.enterprise_roots.enabled` property in the list and set it to **true**.
- If your computer is connected to the network via a proxy server:
 1. Select **Tools > Options** from LoadComplete main menu.
 2. Select **Network > Proxy** from the tree in the **Options** dialog.
 3. Select *Manual Proxy Configuration* from the **Mode** drop-down list.
 4. Enter the address and port of your proxy server. If the proxy server requires authorization, enter your login and password.
 5. Click **OK** to save the changes and close the dialog.

- (Optional) You can add a server, where your sample application resides, to the **Record traffic to these hosts only** list:
 1. In LoadComplete, select **Tools > Options** from the main menu. LoadComplete will open the Options dialog.
 2. Choose **Recording > Filters** from the tree in the left part of the dialog.
 3. On the Record traffic to these hosts only list, enter the name of the computer, where the sample application resides. If it resides on the same computer, where LoadComplete is installed, enter *localhost* or the network name of that computer.

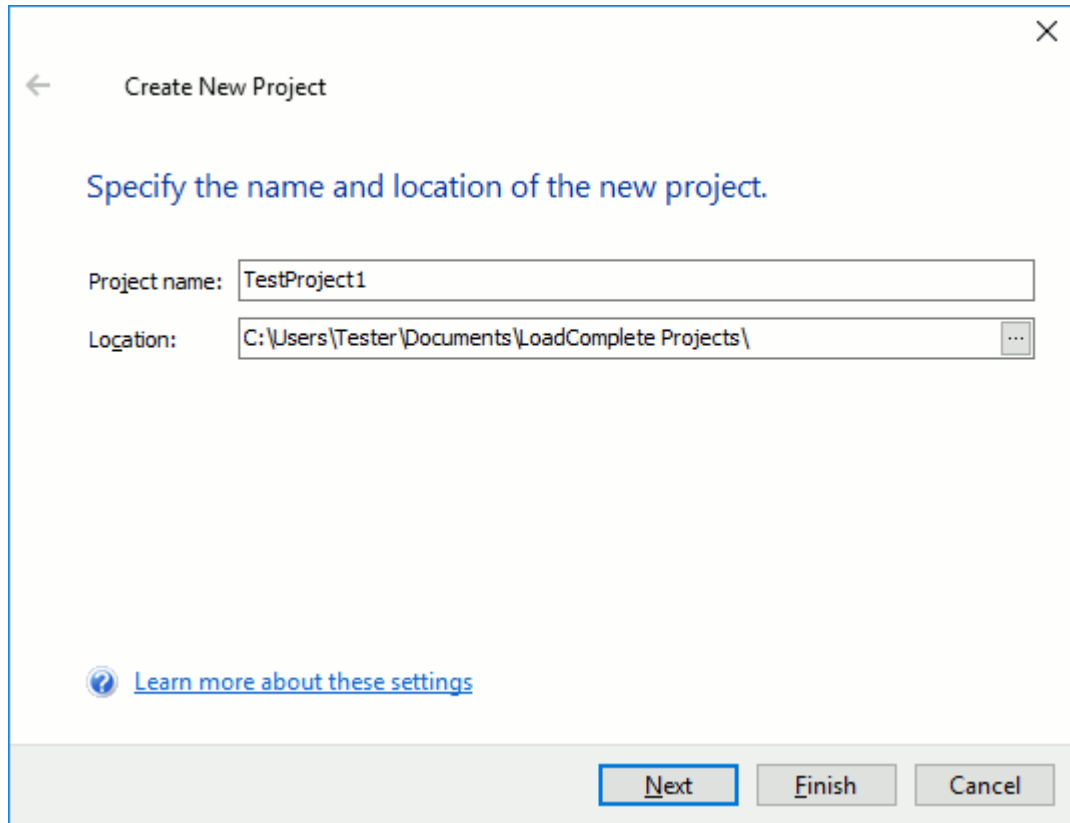
LoadComplete will ignore requests other applications on your computer may send to network during the test recording. This way, you will be able to avoid unnecessary traffic in your scenario.

For more information on preparing to record traffic, see *Preparing for Recording* in LoadComplete documentation.

1. Creating a Test Project

1. If you have a project opened in LoadComplete, close it. To do this, choose **File | Close** from LoadComplete main menu.

2. Select **File | New | New Project** from LoadComplete main menu. LoadComplete will open the **Create New Project** wizard.



← Create New Project

Specify the name and location of the new project.

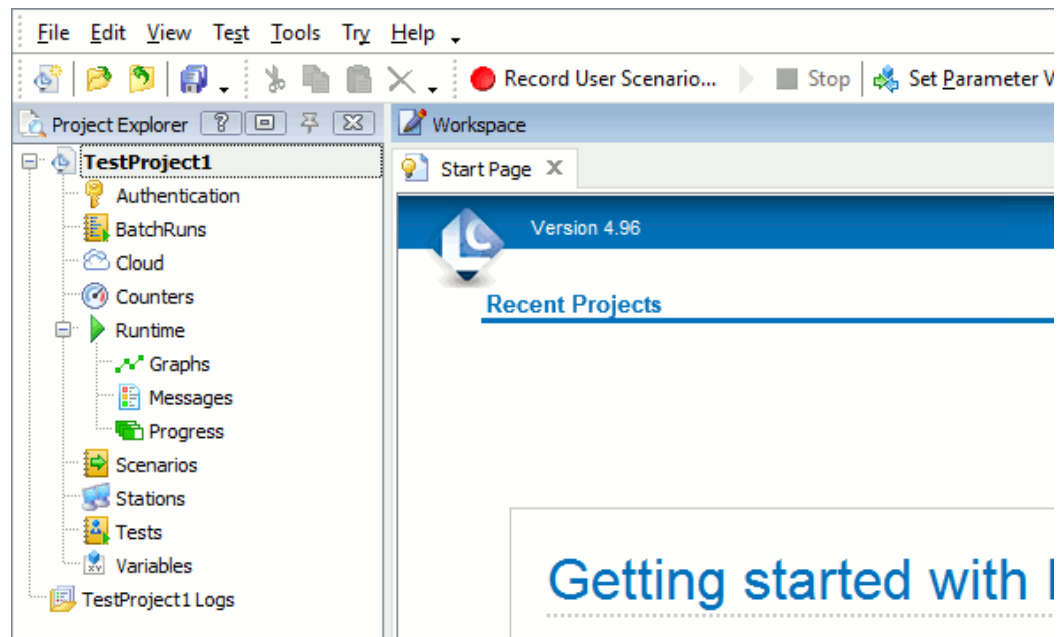
Project name:

Location:

[Learn more about these settings](#)

3. In the dialog, enter the project name into the **Project name** edit box. LoadComplete will automatically generate the path for the project file and display it in the **Location** field. The project folder is used to store all information generated for or by the project: scenarios, tests, test log files and so on. You can change the project's folder in the Location box.
4. On the next page of the wizard, you can enable the performance monitoring for your tested web server. This tutorial does not cover the performance monitoring, therefore, you can click **Finish** to complete the project creation.

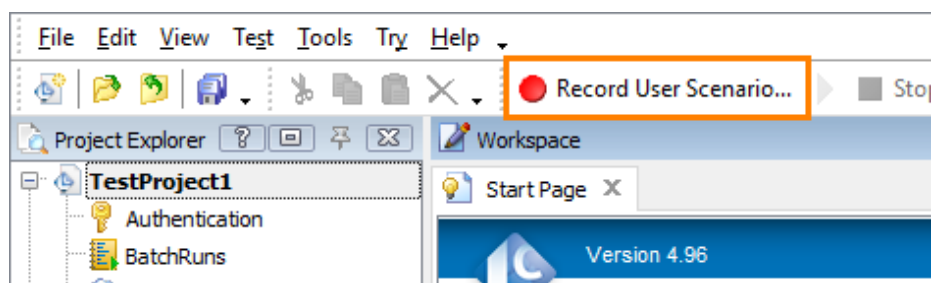
LoadComplete will create the project and display its contents in the Project Explorer panel:



2. Recording a User Scenario

After you have created a test project, you can record scenarios to be simulated by one or more virtual users. To create a simple website load test, you need to record at least one user scenario. To do this:

1. Click **Record User Scenario** on LoadComplete Test Engine toolbar:



2. LoadComplete will open the following dialog:

3. In the dialog:

- Enter the desired scenario name.
- Enter the name of the single-user test to be recorded.

(The single-user test is a test that LoadComplete automatically creates when recording a user scenario. Typically, single-user tests are used to verify that the scenario works as expected. See the next section *Verify the Recorded Scenario* for more information. You can also use the single-user tests to create load tests on basis of the recorded scenarios.)

- Specify whether LoadComplete will automatically launch a web browser during scenario recording. For this purpose, enable the **Launch web browser** check box.
- Select the web browser to use for test recording.

The dialog lists browsers installed in your operating system. If you are going to use a non-standard browser (for instance, if you are going to use a custom client application), clear the **Launch web browser** check box.

- In the **Cache mode** list, select **Private mode** to activate the web browser's private mode before recording.
- Make sure the Record traffic from this browser only setting is enabled. This will filter out unneeded traffic from other browsers and applications.

Click **Record** to start recording.

Upon closing the dialog, LoadComplete checks whether your recording settings match the

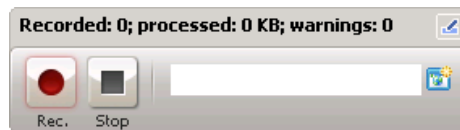
settings recommended for recording scenarios without unneeded traffic. If they differ, it displays a message informing you about this and suggesting that you set the recommended values and continue. Of course, you can start recording with your current settings, if needed.

4. LoadComplete will configure your operating system to connect to the network through the LoadComplete proxy. (LoadComplete will restore your system settings after the recording is over).

If you use Firefox for recording, and your Firefox browser is not configured to use system proxy settings to connect to the network, LoadComplete will show you a warning. Configure Firefox to use system proxy settings and click **Continue** to proceed with recording.

If you use Firefox for recording, and your Firefox browser is not configured to use a Windows certificate store, LoadComplete will show you a warning. In this tutorial, you can ignore the warning and click **Continue** to proceed with recording right away. However, we recommend that you configure your Firefox browser to use Windows certificate store to avoid possible issues with recording secured traffic.

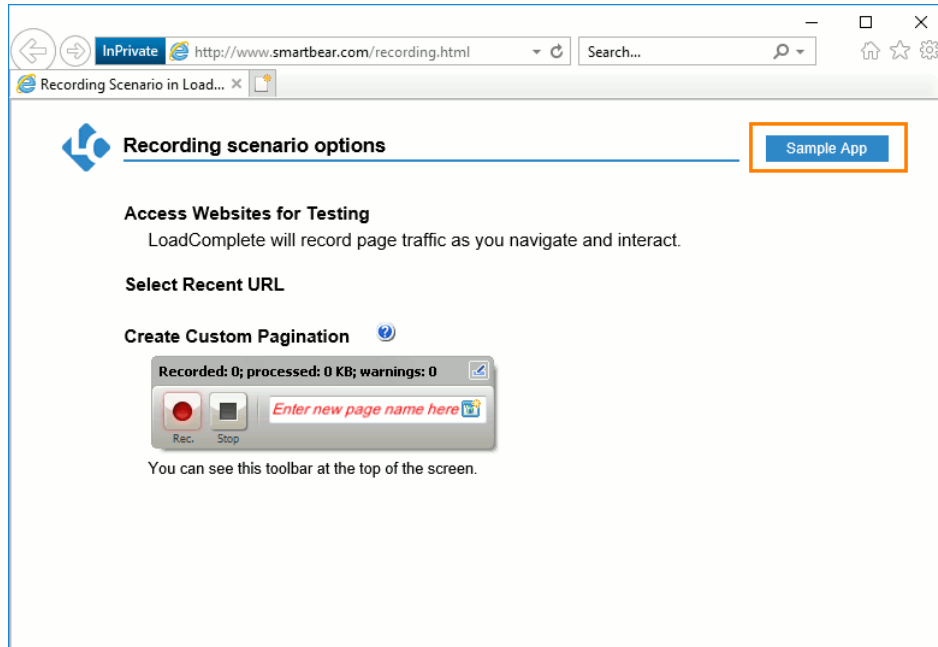
5. After LoadComplete has started recording, it will display the *Recording* toolbar.



6. LoadComplete will also launch the web browser you have selected in the **Record User Scenario** dialog and open the **Record Scenario Options** web page. (This page resides on your computer and LoadComplete will not record opening it.)

From this page, you can navigate to a needed web page.

Click **Sample App** to navigate to the sample Web Orders application:



If you have not chosen a web browser for LoadComplete to launch, launch your browser and open the <http://localhost/LoadComplete> web page manually.

Note: If you are unable to open this page, make sure that both the sample app and IIS are installed on your computer, and IIS includes the features needed for testing (.NET Extensibility, ASP.NET 3.5, ISAPI Extensions, and ISAPI Filters). For more information, see the LoadComplete Basic Test Sample description in LoadComplete documentation.

7. Perform the following actions (this is the sample sequence of expected actions):

Note: As you perform those actions, LoadComplete will process them and add requests to the scenario. The Recording toolbar will show the number of recorded requests, the amount of request and response data processed, and the number of warnings detected.

- Log in to the web application using the data the Login page contains:

Username: Tester

Password: test

Click **Login**.

The application will display the main page that contains a list of orders. On the main page, you can create, modify or delete orders.

- After the main page is displayed, you can perform the desired actions against the application (LoadComplete will simulate these actions for virtual users during the test run).

In this tutorial, we will create a new order, make sure the application has added it to the list of orders and then delete the order.

To create a new order, click the **Order** link on the left.

The screenshot shows a web browser window with the URL `http://localhost/LoadComplete/default.aspx`. The page title is "Web Orders" and it displays a "List of All Orders" table. The sidebar on the left contains three links: "> View all orders", "> View all products", and "> Order" (which is highlighted with an orange border). The table below has the following data:

	Name	Product	#	Date	Street	City	State	Zip	Card	Card Number	Exp	
<input type="checkbox"/>	Paul Brown	ScreenSaver	2	03/12/2010	5, Ringer Street	Las Vegas, NV	US	748	MasterCard	123456789012	02/07	
<input type="checkbox"/>	Mark Smith	FamilyAlbum	1	03/07/2010	9, Maple Valley	Whitestone, British Columbia	Canada	76743	VISA	770077007700	01/09	
<input type="checkbox"/>	Steve Johns	ScreenSaver	1	02/26/2010	17, Park Avenue	Salmon Island	Canada	21233	MasterCard	444555444555	03/09	
<input type="checkbox"/>	Charles Dodgeson	MyMoney	1	02/15/2010	45, Stone st.	Bringtone, TX	US	23233	American Express	333222333222	07/10	
<input type="checkbox"/>	Susan McLaren	MyMoney	1	01/05/2010	7, Flower Street	Earlcastle	Great Britain	21444	MasterCard	999888777888	04/10	
<input type="checkbox"/>	Bob Feather	FamilyAlbum	1	12/31/2009	14, North av.	Milltown, WI	US	81734	VISA	111222111222	06/08	
<input type="checkbox"/>	Samuel Clemens	MyMoney	2	12/21/2009	3, Garden st.	Hillsberry, UT	US	53665	MasterCard	398743242342	03/09	
<input type="checkbox"/>	Clare Jefferson	FamilyAlbum	2	12/04/2009	23, Owk Street	Greentown, CA	US	63325	MasterCard	770000770000	03/08	

Below the table is a "Delete Selected" button.

The Order link

After that the browser will switch to the page that lets you add and modify orders.

- To create an order, follow the instructions below:

Product: select *FamilyAlbum* from the drop-down list.

Quantity: enter *50* as the value of this field.

Click **Calculate** to populate the field connected with the price automatically.

Customer name: enter *John Smith Jr.* as the value of this field.

Street: enter *Orange Blvd.* as the value of this field.

City: enter *Grovetown, CA* as the value of this field.

State: enter *US* as the value of this field.

Zip: enter *13457* as the value of this field.

Select **MasterCard**.

Card Nr: enter *111222333444* as the value of this field.

Expire date (mm/yy): enter *05/12* as the value of this field.

The Order Window

- Click **Process** to add the order with the specified information to the orders table.
- Switch to the orders table. To do this, click the **View all orders** link on the left.

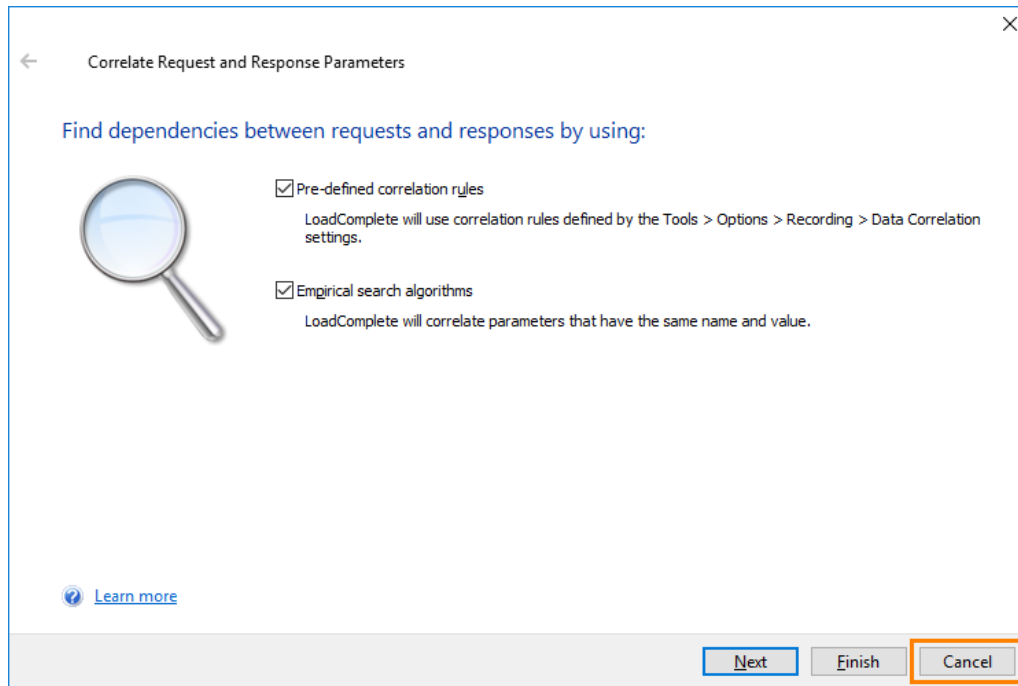
Make sure that the created order is shown in the table:

	Name	Product	#	Date	Street	City	State	Zip	Card	Card Number	Exp	
<input type="checkbox"/>	John Smith Jr.	FamilyAlbum	50	01/19/2018	Orange Blvd.	Grovetown, CA	US	13457	MasterCard	111222333444	05/12	
<input type="checkbox"/>	Paul Brown	ScreenSaver	2	03/12/2010	5, Ringer Street	Las Vegas, NV	US	748	MasterCard	123456789012	02/07	
<input type="checkbox"/>	Mark Smith	FamilyAlbum	1	03/07/2010	9, Maple Valley	Whitstone, British Columbia	Canada	76743	VISA	770077007700	01/09	
<input type="checkbox"/>	Steve Johns	ScreenSaver	1	02/26/2010	17, Park Avenue	Salmon Island	Canada	21233	MasterCard	444555444555	03/09	
<input type="checkbox"/>	Charles Dodgeson	MyMoney	1	02/15/2010	45, Stone st.	Bringtone, TX	US	23233	American Express	333222333222	07/10	
<input type="checkbox"/>	Susan McLaren	MyMoney	1	01/05/2010	7, Flower Street	Earicastle	Great Britain	21444	MasterCard	999888777888	04/10	
<input type="checkbox"/>	Bob Feather	FamilyAlbum	1	12/31/2009	14, North av.	Miltown, WI	US	81734	VISA	111222111222	06/08	
<input type="checkbox"/>	Samuel Clemens	MyMoney	2	12/21/2009	3, Garden st.	Hillsberry, UT	US	53665	MasterCard	398743242342	03/09	
<input type="checkbox"/>	Clare Jefferson	FamilyAlbum	2	12/04/2009	23, Owk Street	Greentown, CA	US	63325	MasterCard	770000770000	03/08	

The new order

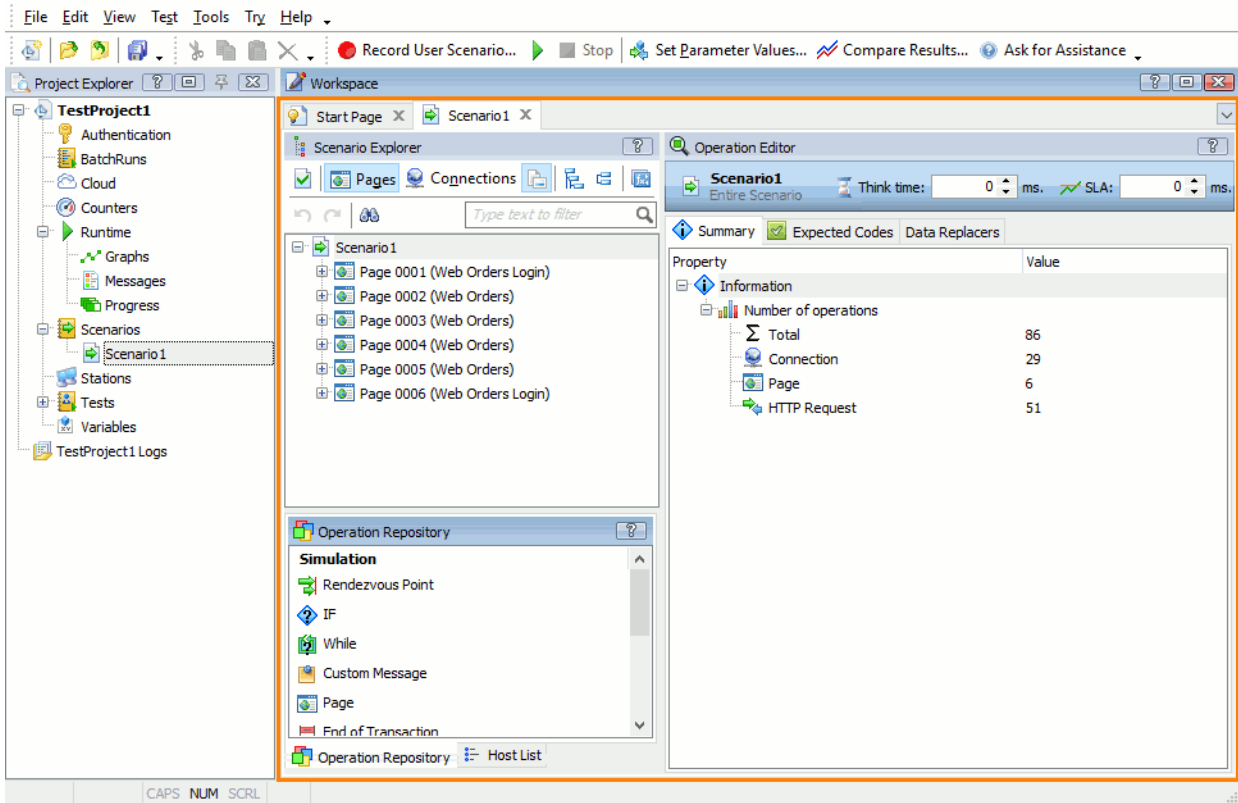
- To delete the created order, select the check box to the left of the order and then click **Delete Selected**.
- Now we will close the browser (click the X button at its caption).

8. After you have performed all the desired actions against the web site, click **Stop** on the Recording toolbar. This will stop the recording.
9. LoadComplete will show the **Correlate Request and Response Parameters** wizard suggesting that you find dynamic parameters in the recorded scenario and correlate them. Data correlation is beyond this tutorial scope. In the dialog, click **Cancel** to skip the data correlation.



3. Viewing the Recorded Scenario

You can examine the recorded traffic both during the recording, and after you finish it. To open the recorded scenario, double-click the scenario in the **Project Explorer**. LoadComplete displays the scenario in the **Scenario** editor:



Recorded Scenario


The scenario consists of requests sent to the test server during the recording. You can select any request and examine it on the **Request** page.

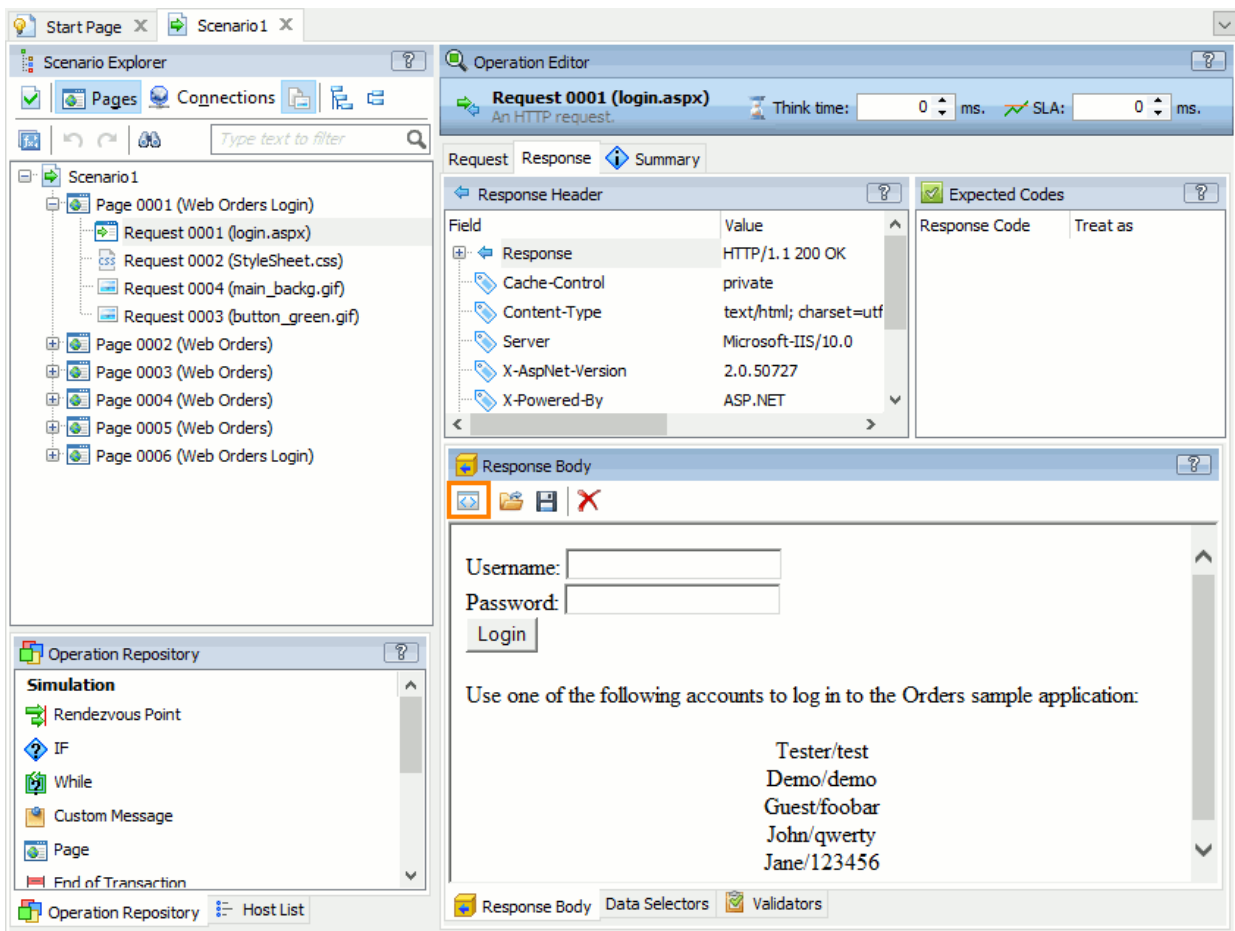
For a POST request, you can view and edit parameters passed to the tested web server:

The screenshot displays the SmartBear LoadComplete interface. The 'Scenario Explorer' on the left shows a tree view of a scenario named 'Scenario1', with 'Request 0027 (Process.aspx)' selected. The 'Operation Editor' on the right shows the details for this request. The 'Request Header' section lists several headers: 'Request' (POST /LoadComplete/Process.aspx HTTP/1.1), 'Accept' (text/html, application/xhtml+xml, image/jxr, */*), 'Referer' (http://localhost/LoadComplete/Process.aspx), and 'Accept-Language' (en-US). The 'Request Body' section, which is highlighted with an orange border, shows a list of parameters in a table format:

Field	Value
ctl00%24MainContent%24f...	50
ctl00%24MainContent%24f...	80
ctl00%24MainContent%24f...	15
ctl00%24MainContent%24f...	3400
ctl00%24MainContent%24f...	John+Smith+Jr.
ctl00%24MainContent%24f...	Orange+Blvd.
ctl00%24MainContent%24f...	Grovetown%2C+CA
ctl00%24MainContent%24f...	US
ctl00%24MainContent%24f...	13457
ctl00%24MainContent%24f...	MasterCard
ctl00%24MainContent%24f...	111222333444
ctl00%24MainContent%24f...	05%2F12

The parameters of a POST request

The **Response** page shows detailed information on the response obtained for the request. For instance, it contains the source code of the web page opened when the specified request was sent. To view the sample preview of the page, uncheck the  item on the panel's toolbar:

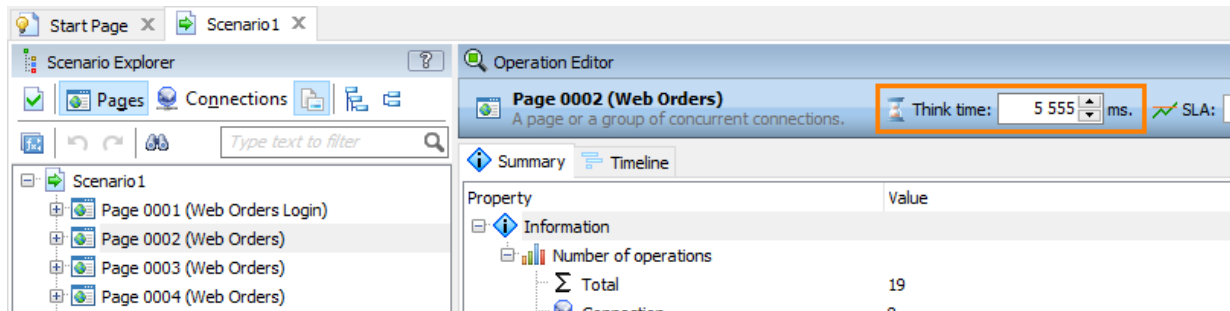


The page preview

- ❗ The preview of the recorded pages can look different from the way these pages look in web browsers. This is because the preview neither loads external resources, such as images or stylesheets, nor executes scripts.

LoadComplete automatically detects concurrent connections that are established by the application to send requests to the server. This makes traffic simulation closer to real life. For more information on working with concurrent requests, see *Configuring Parallel Requests* in LoadComplete documentation.

For each load web page, established connection, and sent request, LoadComplete shows the “think time”.



Think Time

For web pages, for example, the think time is the period passed after the previous web page was loaded completely and before the current page started loading. LoadComplete captures the think time values automatically when you are recording a scenario.

For more information on viewing and editing recorded web traffic in the Scenario editor, see the description of the editor.


4. Verifying Your Recorded Scenario

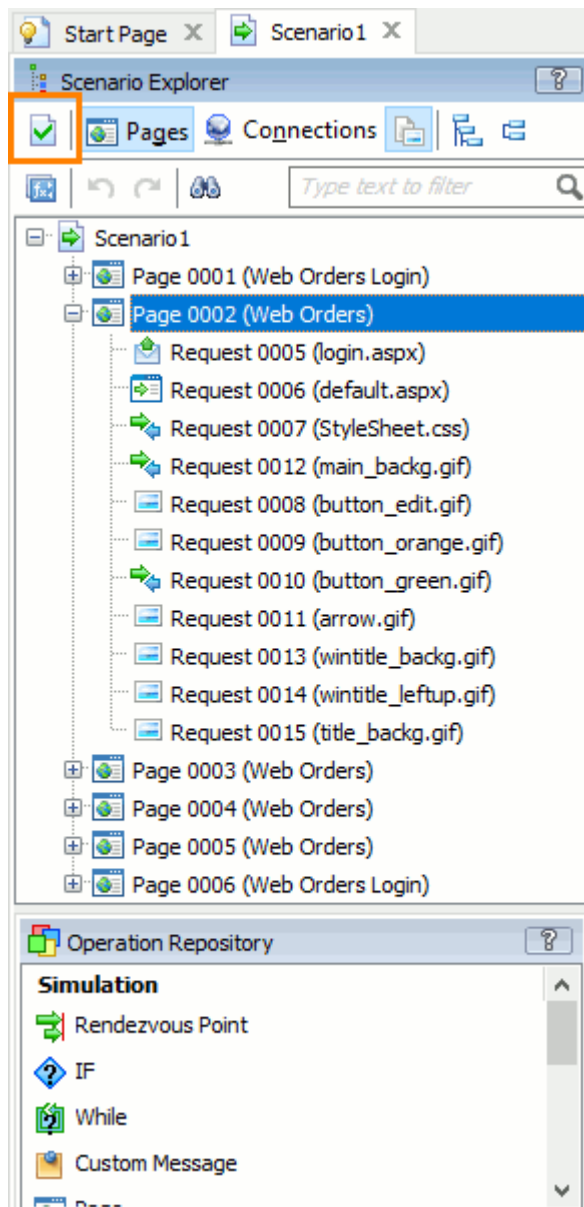
It is a good practice to verify a recorded scenario before creating a website load test. Verification means checking whether LoadComplete plays back the recorded traffic correctly.

There can be two types of verification: initial and advanced. Initial verification means that you run the scenario with only one virtual user. If it works fine for one virtual user, you can play back the scenario with more virtual users. The purpose of advanced verification is to verify whether the scenario works correctly with additional load test settings enabled. In our tutorial, we will use only initial verification.

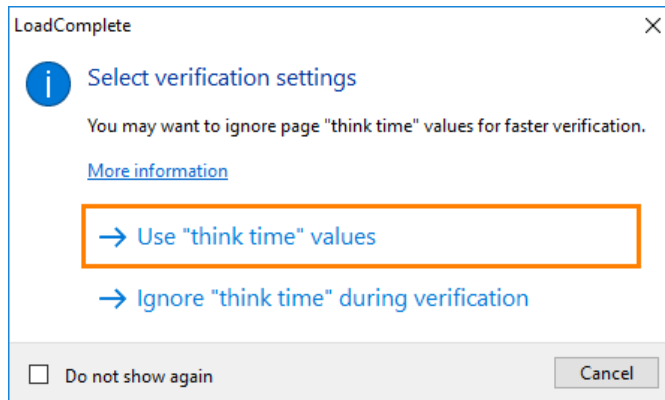
To verify a scenario, you can either run the single-user test, which LoadComplete creates automatically along with recording the scenario, or verify the scenario directly from the **Scenario** editor. We recommend that you verify the scenario from the editor, because it is easier and does not require that the single-user test exist or be assigned for only one virtual user. This way, you can easily verify scenarios every time you record a new scenario or edit an existing one.

To verify a scenario, follow the steps below:

1. Click  **Verify Scenario** on the Scenario editor toolbar.



2. LoadComplete will show the **Select Verification Settings** dialog:




Click **Use “think time” values** to simulate recorded page think time during the verification.

3. LoadComplete will start the verification. It will simulate requests in the recorded scenario for one virtual user. For HTTP requests, it will also compare the actual response codes with recorded response codes.
4. Wait until the run is over.

Note: LoadComplete neither runs the web browser nor simulates user actions by interacting with the tested web site via UI. It just simulates the recorded requests and the entire test run goes invisible.

We recommend that you not browse web pages during the playback as this may affect simulation and may cause the test to fail.

5. After the run is over, LoadComplete will display verification results. LoadComplete considers verification to pass if it has simulated all recorded requests successfully:
 - If LoadComplete fails to simulate a request, it will log an error. There can be various reasons for this, for instance, a network malfunction or an authorization problem.
 - If LoadComplete simulates a request, but the actual response differs from the recorded response, it will log a warning or an error.

If the scenario verification fails, you can click  **Ask for Assistance** at the top of the LoadComplete window to contact a product specialist who can help you fix issues, or you can try troubleshooting the problem yourself. For detailed information on how to do this, see *Resolving Errors and Warnings* in LoadComplete documentation. As an alternative, you can try recording another scenario instead of the one that failed.

If you use LoadComplete Trial, LoadComplete will show a message suggesting that you ask for assistance.

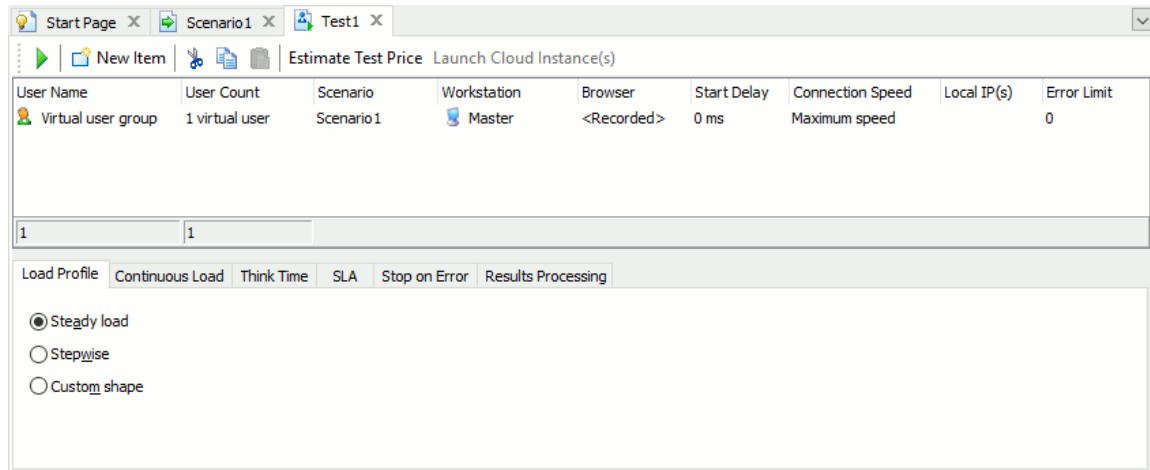
In this tutorial, you can ignore verification warnings and errors. However, in your real-life projects, we recommend that you make sure that the results do not contain any errors and/or warnings.

5. Creating a Load Test

After you have made sure that the scenario works as expected, you can create a load test that will use it. You can create a new test or modify the single-user test by adding more virtual users to it. We will modify the single-user test as it is easier and does not require adding new items to the project.

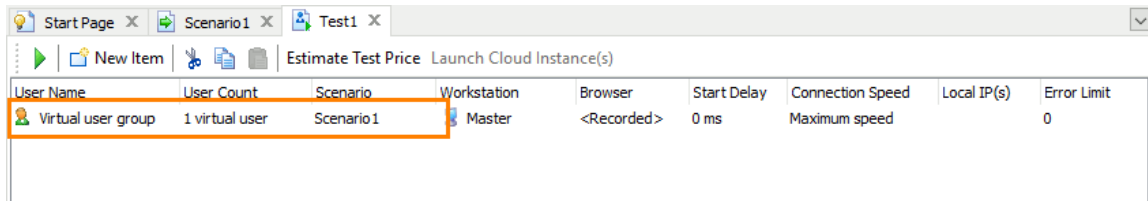
To create a load test from the single-user test:

1. Double-click the test name in the **Project Explorer** panel. This will open the test in the **Load Test Editor**:



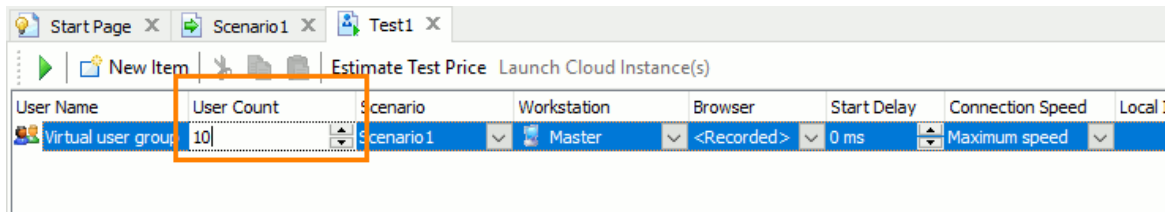
2. The editor shows a table of virtual users in the load test. Each row in this table corresponds to one or more virtual users.

By default, the test contains a single group with one virtual user in it. The user simulates the scenario we have recorded:



User Name	User Count	Scenario	Workstation	Browser	Start Delay	Connection Speed	Local IP(s)	Error Limit
Virtual user group	1 virtual user	Scenario1	Master	<Recorded>	0 ms	Maximum speed		0

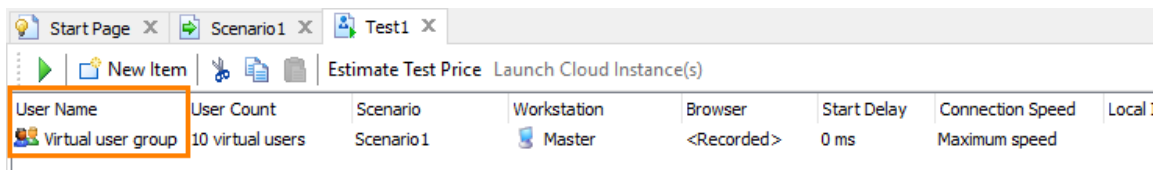
Let's add more virtual users to the group. Enter the number of virtual users in the **User Count** column:



User Name	User Count	Scenario	Workstation	Browser	Start Delay	Connection Speed	Local IP(s)	Error Limit
Virtual user group	10	Scenario1	Master	<Recorded>	0 ms	Maximum speed		

You also can add more virtual users groups to the test. Having multiple user groups can be useful when simulating virtual users with various settings, such as a web browser they use, their connection speed and so on. To identify virtual user groups in test results, the **User Name** is used:

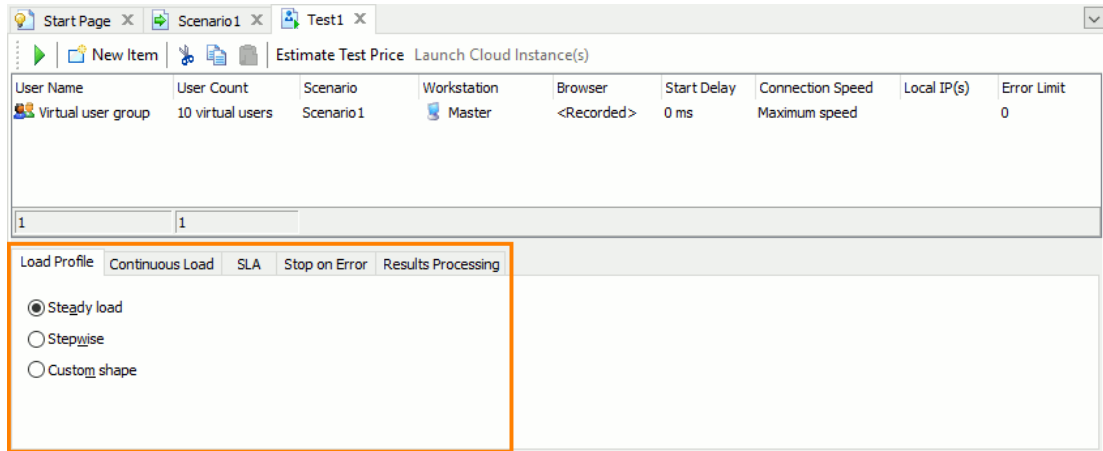
You can also add more virtual user groups to the test. Having multiple user groups can be useful when simulating virtual users with various settings, such as the web browser they use, their connection speed and so on. To identify virtual user groups in test results, use the values specified in the **User Name** column:



User Name	User Count	Scenario	Workstation	Browser	Start Delay	Connection Speed	Local IP(s)	Error Limit
Virtual user group	10 virtual users	Scenario1	Master	<Recorded>	0 ms	Maximum speed		

Note: The total number of virtual users in all the groups in a test can be **no more than 2000**. If you need to simulate more virtual users, you need to use additional machines. Using load stations is beyond this tutorial's scope though. For information on using load stations, see *Distributed Testing* in LoadComplete documentation.


4. At the bottom of the editor there are several tabbed pages that specify additional test parameters:
 - **Load Profile** – Sets the workload profile for virtual users: steady, stepwise ramp-up or custom.
 - **Continuous Load** – Sets the overall duration for the load test and this way runs multiple test iterations if needed.
 - **Think Time** – Sets the think time (the wait time between web page requests) of virtual users.
 - **SLA** – Sets service level agreement (SLA) criteria for the given load test.
 - **Stop on Error** – Sets the number of errors, on which LoadComplete will stop the load test run.
 - **Results Processing** – Enables sending test results after the test run is over and sets email addresses to which the results will be sent.

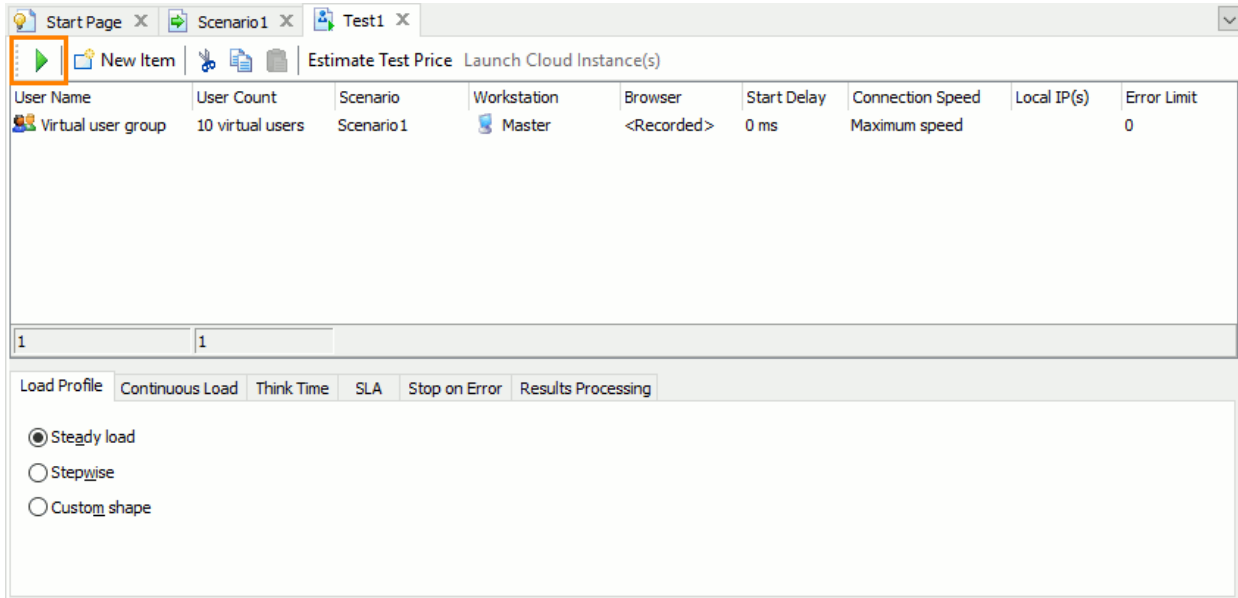


You use them to make your load tests more realistic and closer to real-life situations (see *Emulating Real-Life Conditions* in LoadComplete documentation). For information on using these settings, see *Creating and Configuring Load Tests*.

5. Save the changes.

6. Running Your Load Test

After you are done with your load test, you can run it. Click  **Run This Load Test** on the test editor's toolbar:



If your test simulates more than several hundred virtual users and LoadComplete is configured to gather detailed information on all simulated virtual users or on failed user iterations, LoadComplete will show a warning informing you about possible performance issues. We recommend that you disable gathering detailed information if you do not need it. To do it, click **Reduce logging level and run**. LoadComplete will generate only the Report log after the test run is over. Otherwise, click **Run with current logging settings** to gather all the data.

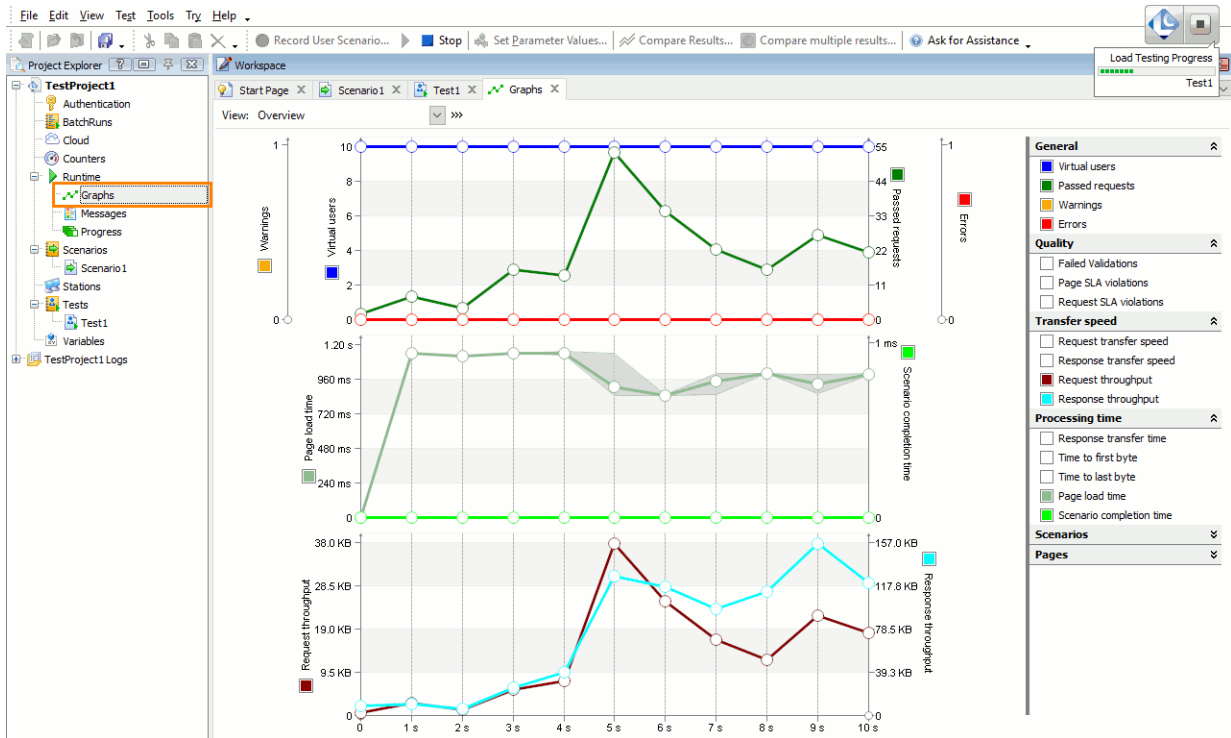
If LoadComplete is configured to store the test log, LoadComplete will check whether the disk where your project is located has enough free space to store the test log (the minimum required amount is 100 MB). If there is not enough space, LoadComplete will show a warning informing you about the issue and telling you how much space you need to free to save the test log successfully. Click **Ignore** to run the test anyway (not recommended). Click **Cancel** to stop the test run, free some space and run the test again.

LoadComplete will start the test run.

Please bear in mind that the test run goes invisible. We do not recommend that you browse web pages during the run as this may affect the traffic simulation and make the test fail.

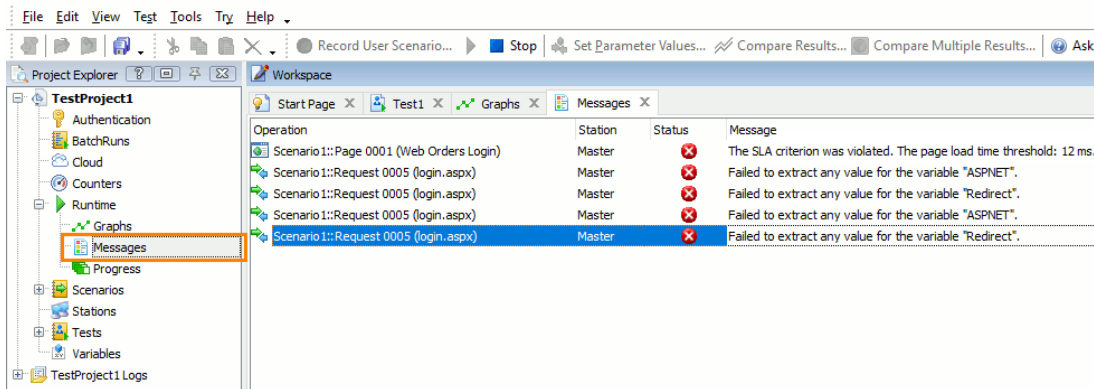
You monitor your load test run on the **Runtime** pages of your project.

The **Runtime > Graphs** page shows the number of simulated virtual users, the number of errors and warnings occurred during the test run, the number of requests simulated successfully, and other performance metrics. Below is a sample view of the page:

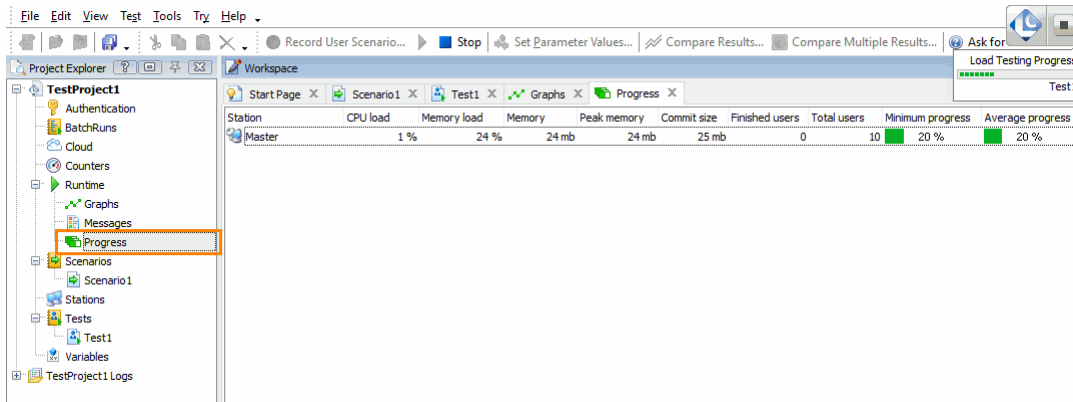


The Graphs page

The **Runtime > Messages** page shows the messages that were displayed during the test run. By default, it shows only errors. Below is a sample view of the page:



The **Runtime > Progress** page shows the run progress for each group of virtual users of your test:



The Progress page

Note: The Runtime pages show statistics only for the current run. As soon as you run your load test again, the previous data will be removed from the pages. You can view the data in the test log.

After the test run is over, LoadComplete will generate a test log and open it for viewing. For information on analyzing test results, see the next step of this tutorial.

If test results include error or warning messages, you can click **Ask for Assistance** at the top of the LoadComplete window to contact a product specialist who can help you fix issues or you can try troubleshooting the problem yourself (see *Resolving Errors and Warnings*).

If you use LoadComplete Trial, LoadComplete will show a message suggesting that you ask for assistance.

In this tutorial, you can ignore test errors and warnings and proceed to the next step.

7. Analyzing Results

After a load test run is over, LoadComplete automatically shows the **test log**. It contains information on the test run, executed requests and received response codes. We use this data both to analyze load test results and find possible causes of a test failure.

Report Panel

The **Report** panel shows summary information on the load test. It also contains various graphs, charts and tables that show load testing statistics.


LoadComplete automatically opens the Report page after the load test is over. You can also open the page by double-clicking the **Report** item in the **Project Explorer**. For more information on the panel, its tabbed pages and information they show, see the description of the panel in LoadComplete help.

Details Panel

The **Details** panel shows information on all the requests simulated during the test run and the responses received from the server. For more information, see the description of the panel in LoadComplete help.

If no requests were simulated with warnings or errors, LoadComplete considers the test run successful. Otherwise, it considers the test failed.

Resolving Issues

If you need help with analyzing results and resolving errors and warnings, click  **Ask for Assistance** on the Test Engine toolbar at the top of LoadComplete's window. This will open a web page, where you can contact product specialists. You can also try troubleshooting yourself. See *Resolving Errors and Warnings*.

If you use LoadComplete Trial, LoadComplete will suggest contacting a product specialist after scenario verification fails, or if a load test with 25 or fewer virtual users fails.

Where to Go Next

This concludes the Creating Your First Test tutorial. We hope it has helped you to get acquainted with LoadComplete. To get more information about LoadComplete and its features, please refer to the LoadComplete Help. Below is a list of help topics that may be of interest to you.

⇒ *Creating Load Tests*

This section provides information on how to create load tests in LoadComplete and to perform typical tasks like synchronizing virtual users or defining workload profiles.

⇒ *Recording New Scenario*

This section provides information on recording scenarios in LoadComplete.

⇒ *Typical Use Cases*

This section provides information on how to set up a test procedure and perform typical load

testing tasks with LoadComplete.

⇒ *Parameterizing Requests*

This section describes how you can use external data for parameters of simulated requests.

⇒ *Data Correlation*

This section describes how to extract data from server responses and pass it to further requests so that the simulated traffic is properly correlated.

⇒ *Simulating Virtual Users on Multiple Workstations*

Explains how to share the load among several workstations.

⇒ *Load Testing on Cloud Computers*

Explains how to generate a load from cloud machines and describes the built-in cloud support provided by LoadComplete.

⇒ *Performance Monitoring*

Describes how you can monitor performance of the tested server with LoadComplete.

⇒ *Running Load Tests*

This section provides information on how to run load tests.

⇒ *Analyzing Results*

This section provides information on analyzing load test results.

Index

	A	
Analyzing test results		32
	C	
Creating		11
Load tests		25
Test projects		11
Creating load tests		25
	L	
Load testing		
Basic concepts.....		4
LoadComplete user interface overview		7
	R	
Recording		
Scenarios		13
Recording user scenarios		13
Running load tests		29
	S	
Scenarios		
Recording		13
Verifying		23
Viewing		20
	T	
Tips.....		6
Emulating real-life conditions		6
	V	
Verifying recorded scenarios		23
Viewing		
Recorded scenario		20
Viewing the recorded scenario		20