



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.

© 2015 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 includes the UnzDll.dll and ZipDll.dll libraries that are supplied by Info-Zip. These modules are copyright © 1990-2005 Info-ZIP. They are provided “as is”, without warranty of any kind, expressed or implied. In no event shall Info-ZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising from the use of or inability to use this software.

LoadComplete also 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	5
LoadComplete User Interface	7
CREATING YOUR FIRST TEST	8
Preparing For Creating Your First Test	8
1. Creating a Test Project.....	10
2. Recording a User Scenario.....	11
3. Viewing the Recorded Scenario.....	17
4. Verifying Your Recorded Scenario.....	19
5. Creating a Load Test.....	21
6. Running Your Load Test.....	24
7. Analyzing Results	25
Report Panel.....	26
Details Panel	26
Where to Go Next	26
INDEX.....	28

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. After you have recorded a scenario, LoadComplete automatically saves it and creates a load test based on the scenario.

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. There can be two types of verification: initial and advanced.

Initial verification of a recorded scenario means simulating it for one virtual user and checking whether LoadComplete simulates the recorded traffic correctly. Advanced verification means checking whether the scenario works correctly with additional load test settings enabled.

4. Creating load tests that will simulate recorded traffic.

In this step, you create a load test on the basis of a verified scenario. Note that 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.

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.

Notes: If you simulate traffic on Amazon cloud computers, the maximum number of concurrent virtual users that can be simulated on a single instance (Amazon cloud computer) is 300. Note that you can assign any number of virtual users to an Amazon machine image in your load test. LoadComplete will launch the needed number of instances automatically.

6. Running load tests.

When you run a load test, LoadComplete launches the embedded Remote Agent utility (*RAS.exe*) that simulates virtual users. The utility 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 instance of the Remote Agent utility can simulate a limited number of virtual users. If your load test simulates more virtual users, LoadComplete launches additional instances of the utility. After the testing is over, LoadComplete closes all extra instances.

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).

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). LoadComplete lets you specify not only the number of virtual users to be simulated, but also some additional settings that allow you to emulate “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:

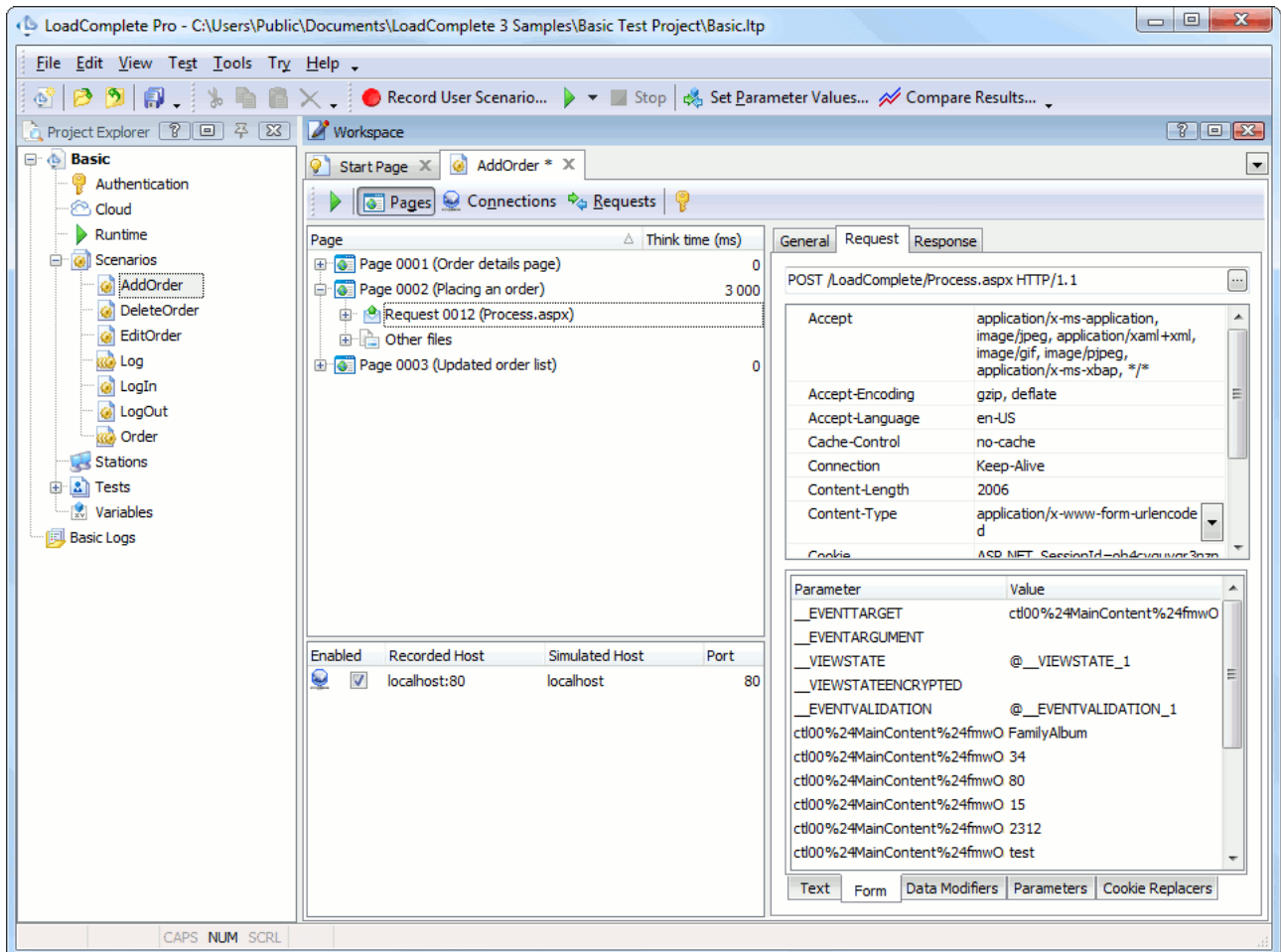
- **Increasing the number of users.** With LoadComplete you can simulate not just a predetermined number of virtual users, but also a growing number that increases every time a specified period expires. This feature is called **load profile** and allows you to specify not only the time period, but also 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 different browsers, the bandwidth of their Internet connection is different and they spend different time on each page. You can configure all these parameters when creating load tests in LoadComplete. You can specify different browsers and connection bandwidth for different groups of virtual users. You can also specify browsing time for each tested page (we call this time **think time**, as it simulates the time period when a user is viewing the page and thinking). You can easily randomize the think time for simulated scenarios to better emulate real user activity. 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.

In this 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) 6.0 or later,
- .NET Framework 2.0 or later,
- Microsoft Data Access (if you need to work with the database files).

Also, in order for the sample to work properly, you may need to enable certain IIS features on some computers. See *Basic Test Sample* in LoadComplete help for complete information.

This tutorial includes the following steps:

Preparing For Creating Your First Test

- 1. Creating a Test Project*
- 2. Recording a User Scenario*
- 3. Viewing the Recorded Scenario*
- 4. Verifying Your Recorded Scenario*
- 5. Creating a Load Test*
- 6. Running Your Load Tests*
- 7. Analyzing Results*


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 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, open the **Advanced** page.
3. Switch to the **Network** tabbed page and click **Settings**.
4. In the **Connect Settings** dialog, select **Use system proxy settings**.
5. Click **OK** to close the dialog and save the changes. Then close the **Options** dialog.

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

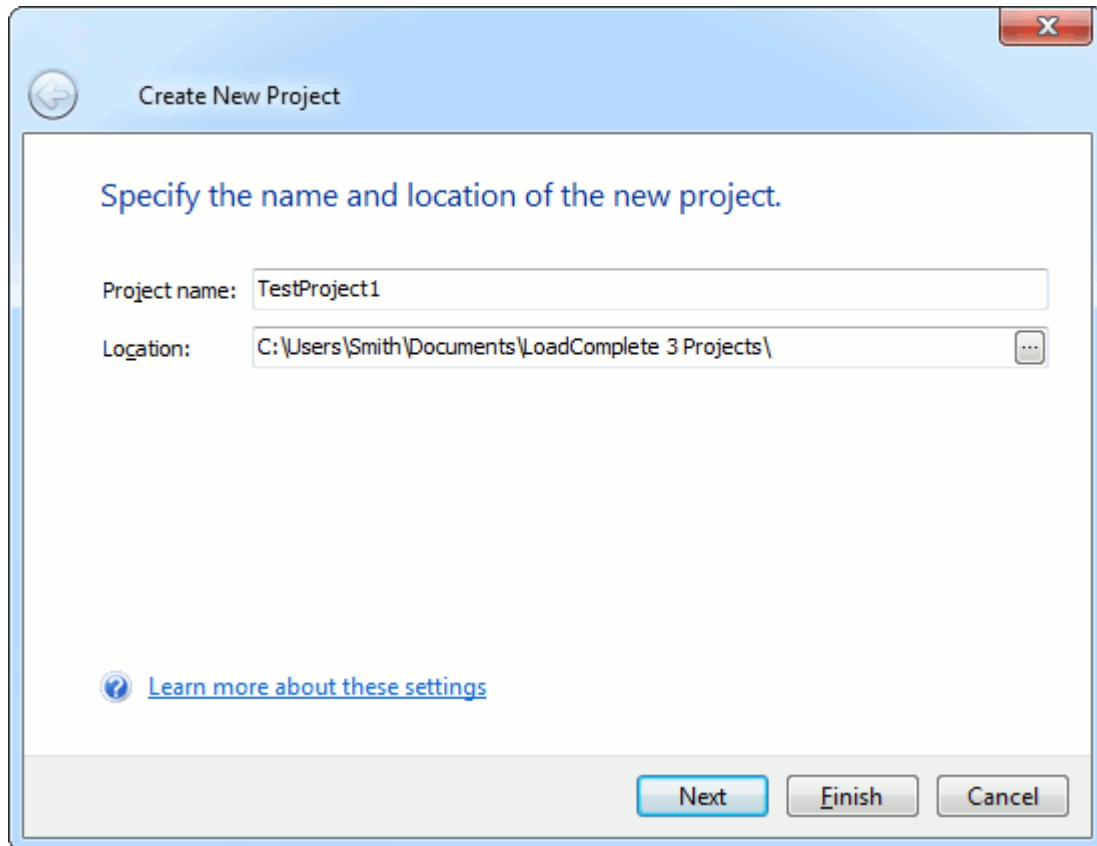
- If your computer is connected to the network via a proxy server:
 1. Select **Tools | Options** from LoadComplete's main menu.
 2. Select **Network | Proxy** from the tree in the **Options** dialog.
 3. Select the **Manual Proxy Configuration** option.
 4. Enter the address and port of your proxy server. 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*.

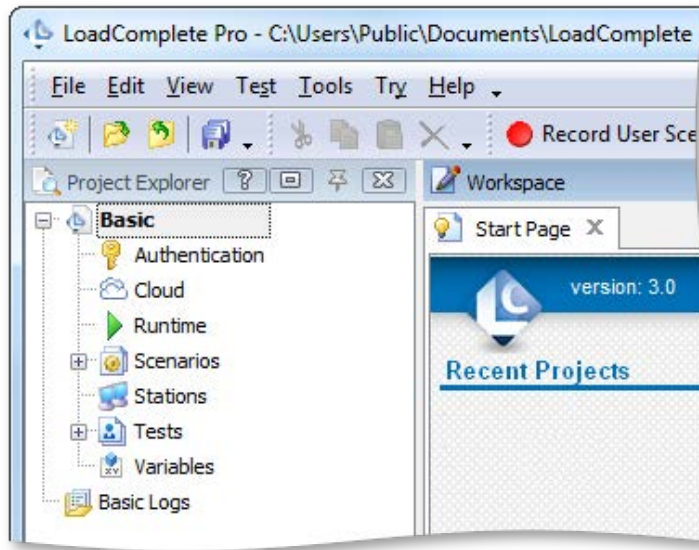
1. Creating a Test Project

1. If you have a project opened in LoadComplete, close it. To do this, choose **File | Close** from LoadComplete's main menu.
2. Select **File | New | New Project** from LoadComplete's main menu. LoadComplete will open the **Create New Project** wizard.




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:



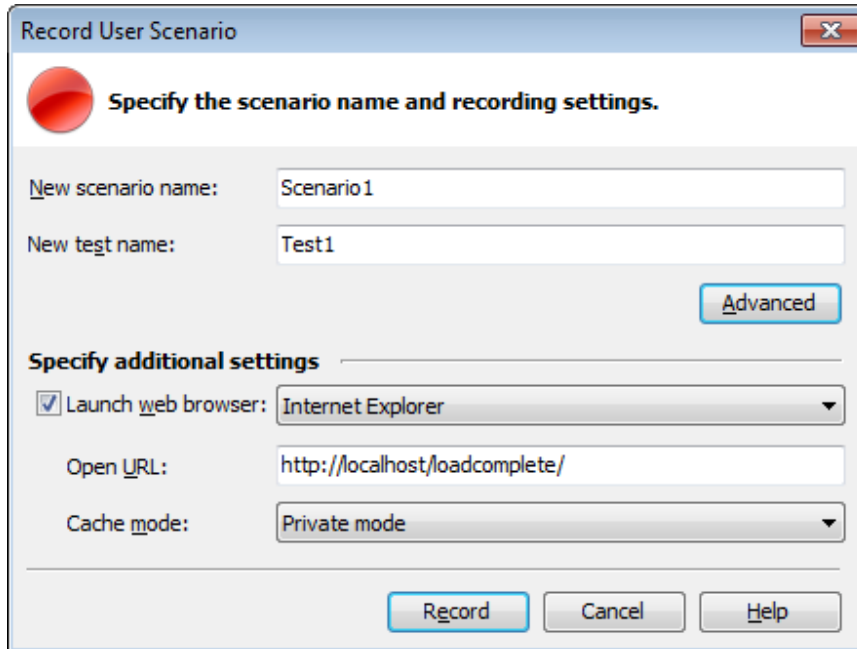
2. Recording a User Scenario

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

1. Click  **Record Load Test** on LoadComplete's Test Engine toolbar:



This will invoke the following dialog:



2. In the dialog:

- Specify 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 the 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.

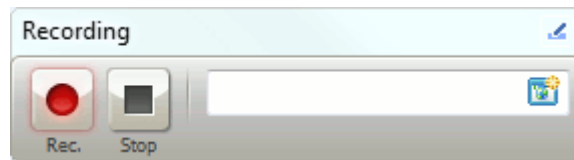
- In the **Open URL** field, type the URL of the web site to be tested:
<http://localhost/LoadComplete/>.
- In the **Cache mode** list, select **Private mode** to activate the web browser's private mode before recording.

Click **Record** to start recording.

3. LoadComplete will configure your operating system to connect to the network through the

LoadComplete proxy and display a message asking you to confirm the changes. Click **Start** to confirm the changes and proceed with recording. (LoadComplete will restore your system settings after the recording is over).

4. If you use Firefox for recording, and it is not configured to use the system proxy settings to connect to a network, LoadComplete displays a warning. Configure Firefox to use the system proxy settings and click **Continue** to proceed with recording.
5. After LoadComplete has started recording, it will display the *Recording* toolbar.



4. LoadComplete will also launch the web browser you selected in the Record User Scenario dialog and open the specified web page. If you have not chosen that option, launch your browser and open the <http://localhost/LoadComplete/> web page manually.
5. Perform the following actions (this is the sample sequence of expected actions):

- 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 application interface for 'Web Orders'. The browser address bar shows 'http://localhost/loadcomplete/default.aspx'. The page title is 'Web Orders' and the user is logged in as 'Tester!'. The main heading is 'List of All Orders'. On the left sidebar, there are navigation links: '> View all orders', '> View all products', and '> Order' (which is highlighted with an orange box). The main content area contains a table of orders with columns: Name, Product, #, Date, Street, City, State, Zip, Card, Card Number, and Exp. Below the table are buttons for 'Check All', 'Uncheck All', and 'Delete Selected'.

	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, Oak Street	Greentown, CA	US	63325	MasterCard	770000770000	03/08	

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 screenshot shows a web browser window with the URL `http://localhost/loadcomplete/Process.a`. The page title is "Web Orders" and the user is logged in as "Tester!". The main content area is titled "Order" and contains three sections:

- Product Information:** Product: FamilyAlbum (dropdown), Quantity: 50, Price per unit: 80, Discount: 15%, Total: 3400. A "Calculate" button is next to the Total field.
- Address Information:** Customer name: John Smith Jr., Street: 12, Orange Blvd, City: Grovetown, State: US, Zip: 13457.
- Payment Information:** Card: MasterCard (selected), Card Nr: 111222333444, Expire date (mm/yy): 05/12.

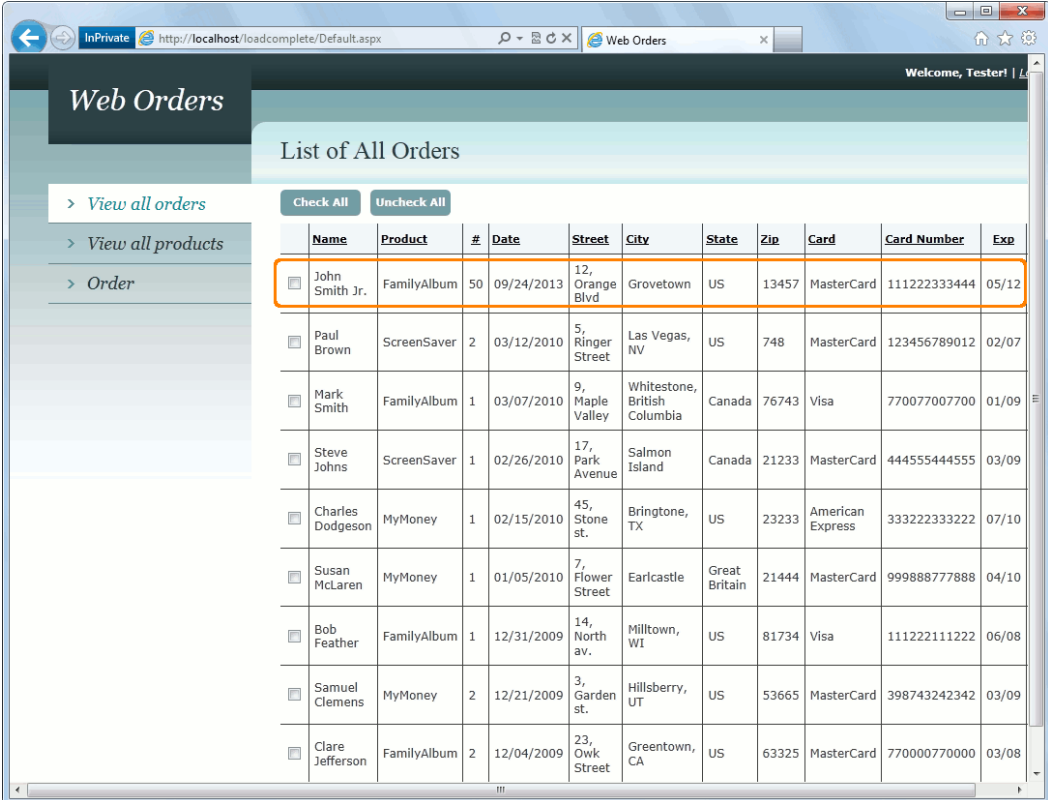
At the bottom of the form are "Process" and "Reset" buttons.

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	09/24/2013	12, Orange Blvd	Grovetown	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	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

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).

LoadComplete will record the requests your browser will send to the server (the Recorded toolbar's caption will show the number of the recorded requests) and the server's responses.

6. After you have performed all the desired actions against the web site, click **Stop** on the Recording toolbar. This will stop the recording.

LoadComplete will process the recorded requests, generate the scenario and display it and the created test in the Workspace panel.

Note: Processing the recorded traffic and generating the scenario may take some time.

3. Viewing the Recorded Scenario

After you have finished the recording, you can examine the recorded traffic. To open the recorded scenario, double-click the scenario in the *Project Explorer*. LoadComplete displays the scenario in the *Workspace* panel.

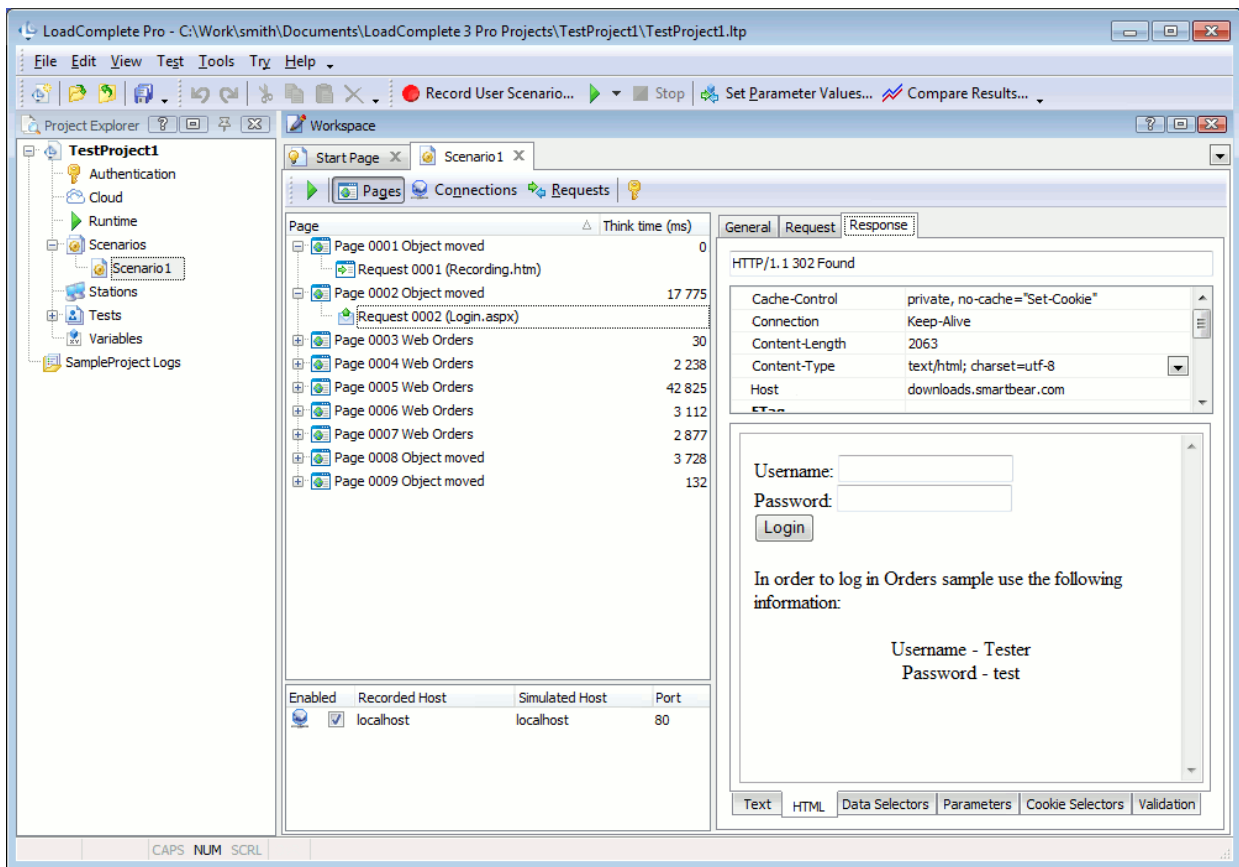
The screenshot shows the LoadComplete interface with the following data:

Page	Think time (ms)
Page 0001 Object moved	0
Page 0002 Web Orders Login	17 775
Request 0002 (Login.aspx)	
Page 0003 Web Orders	30
Page 0004 Web Orders	2 238
Page 0005 Web Orders	42 825
Page 0006 Web Orders	3 112
Page 0007 Web Orders	2 877
Page 0008 Object moved	3 728
Page 0009 Object moved	132

Parameter	Value
__VIEWSTATE	%2FwEPDwUkLTc2NDI1NTk2OGRk%:
__EVENTVALIDATION	%2FwEWBAKz1IqDBQKMwKD1Aglj%
ctl00%24MainContent%24username	Tester
ctl00%24MainContent%24password	test
ctl00%24MainContent%24login_buttc	Login

The scenario contains a list of requests sent to the test server during the recording. You can select any request in the list and examine it more closely in the **Request** tabbed page. For each loaded web page, LoadComplete displays the “think time” — the time period in milliseconds that passed after the last request of the previous page was completed and before the first request of the page was sent. These values are shown in the Think time column. The values are captured automatically when you are recording the scenario. Also, LoadComplete automatically detects concurrent connections that are established by the application to send requests to the server. This information is used to make traffic simulation closer to real life. For more information on simulating concurrent requests with LoadComplete, see *Simulating Parallel Requests* in LoadComplete documentation.

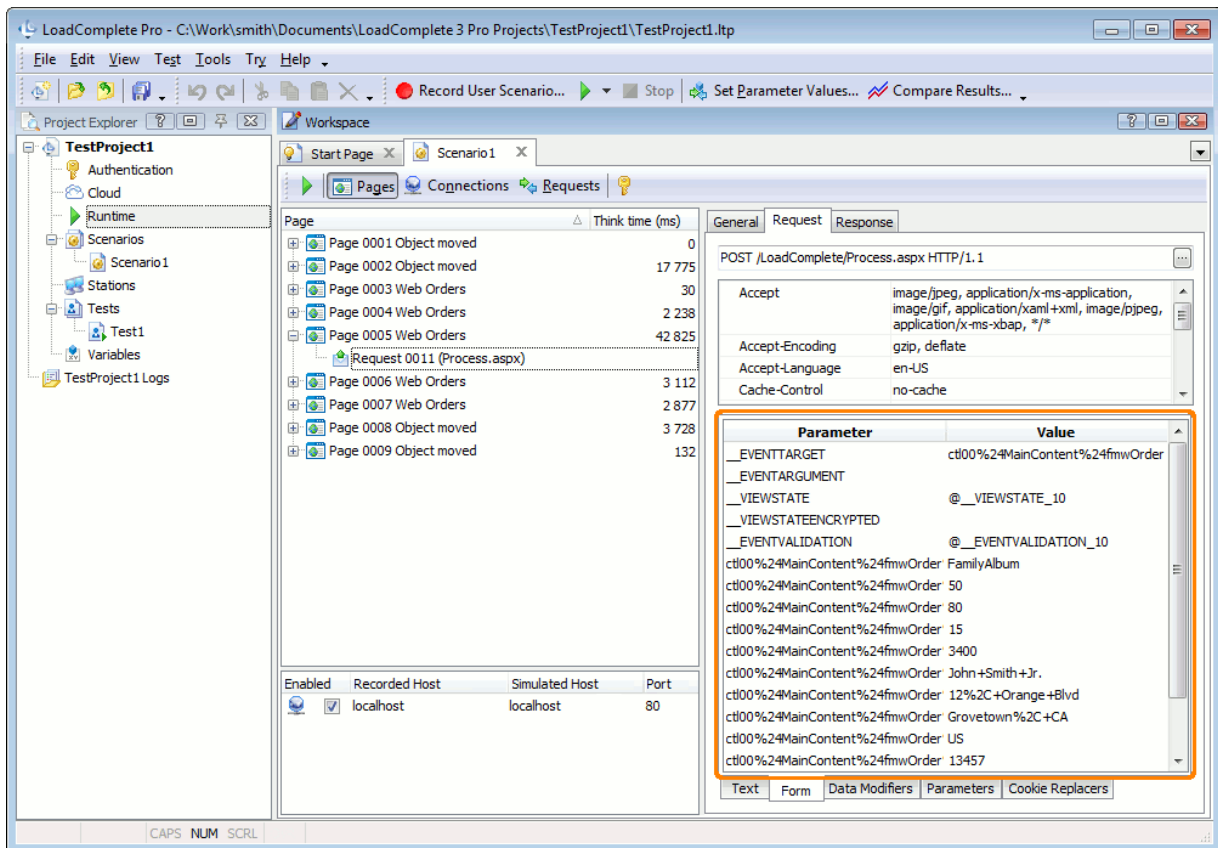
The **Response** tabbed page contains detailed information on the response obtained for the request. For instance, it contains the sample preview of the web page that was opened when the specified request was sent:



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.

The editor also provides you with the detailed information on the parameters that were used for the POST requests. To view these parameters, switch to the **Form** tab of the **Request** page:



The parameters of a POST request

For more information on these pages and information shown in it, see the description of the editor.


4. Verifying Your Recorded Scenario

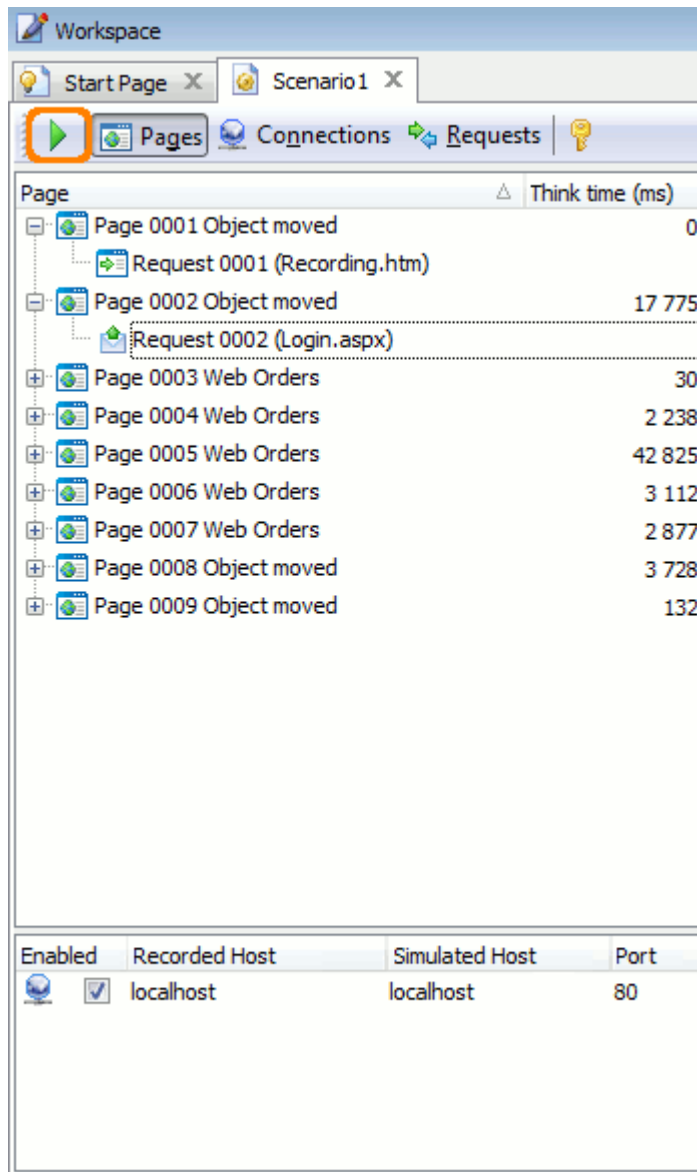
It is a good practice to verify a recorded scenario before creating a website load test. Verification means checking whether LoadComplete plays back the recorded traffic correctly.

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 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.
3. 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.

3. After the run is over, LoadComplete will display test results. Make sure that the results do not

contain any errors and/or warnings.

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.

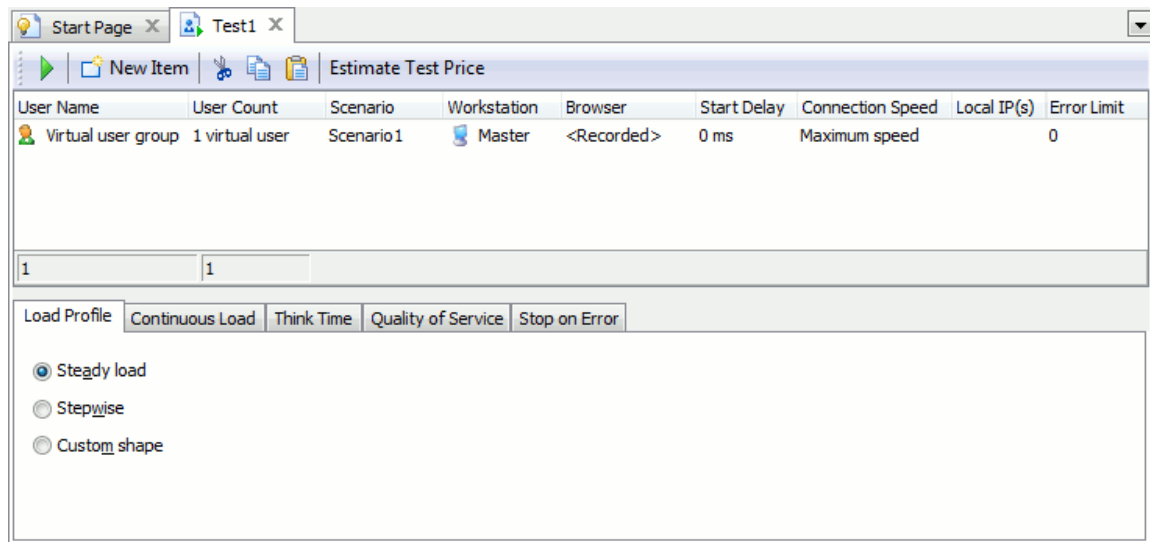
You can try troubleshooting the problem. For detailed information on how to do this, see *Resolving Errors and Warnings*. As an alternative, you can try recording another scenario instead of the one that failed.

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.

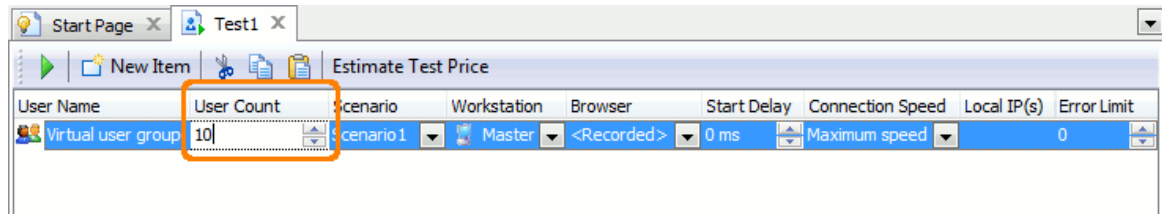
To create a load test from the single-user test:

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

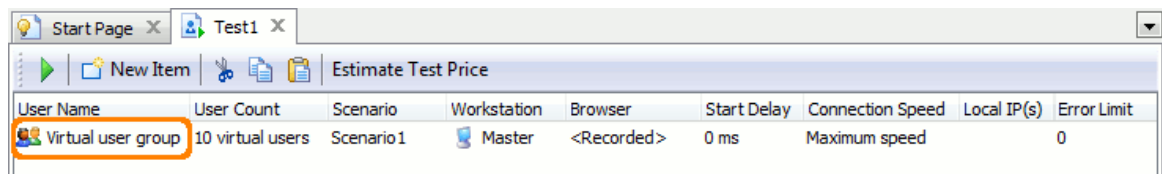


2. The editor shows a table of virtual users in the load test. Each row in this table corresponds to one or more virtual users. Defining multiple virtual user groups can be useful if you want to specify different properties for different user groups, such as web browser, connection speed and so on.

To specify the number of virtual users in a group, enter this value in the **User Count** column:



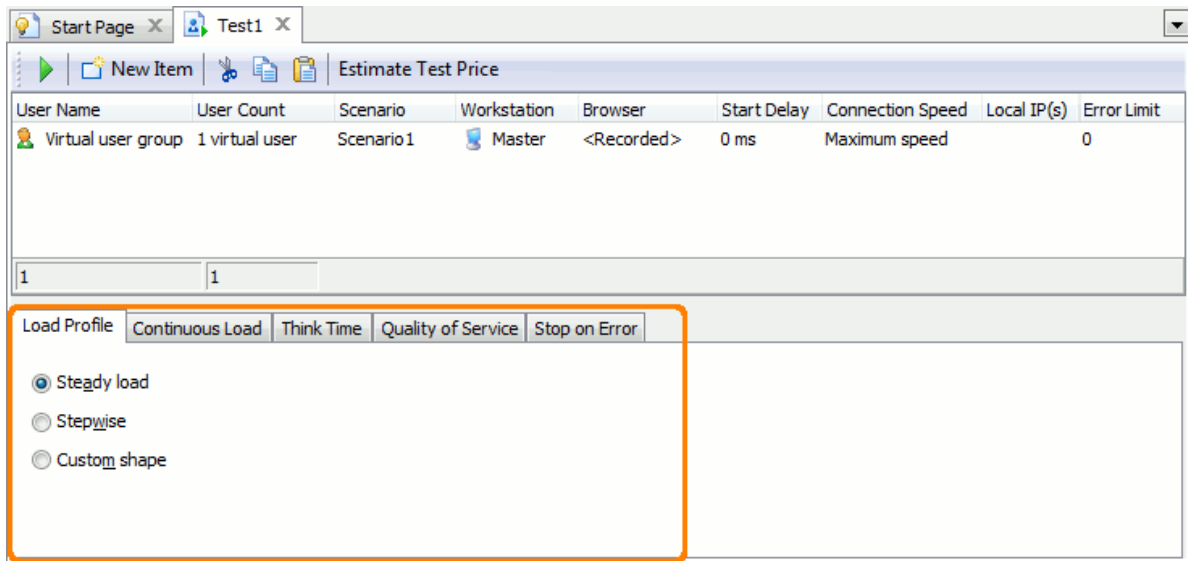
In the **User Name** column, enter the name for the virtual user group. This name will be used to identify users in test results:



3. Add other virtual user groups, if needed.

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.
 - **Random Think Time** – Sets the think time (the wait time between web page requests) of virtual users.
 - **Quality of Service** – Sets the Quality of Service criteria for the given load test.
 - **Stop on Error** – Sets the number of errors, on which LoadComplete will stop the load test run.




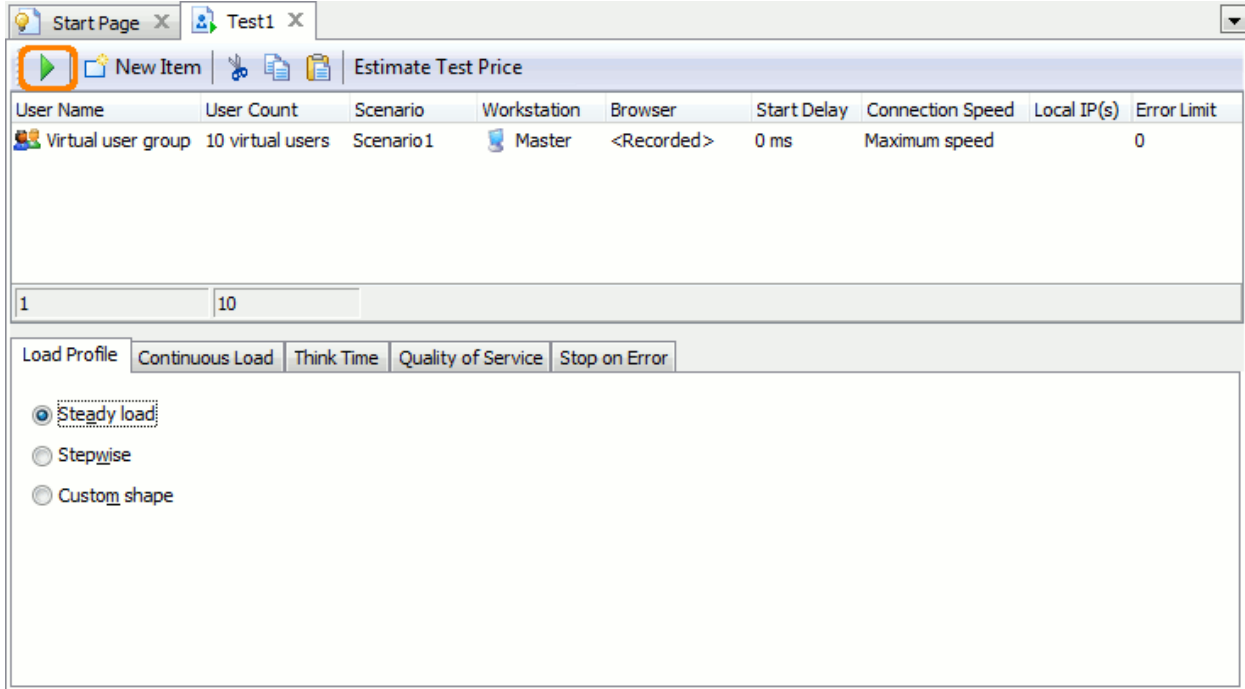
Combining these settings properly, you can make your tests more realistic and closer to real-life situations. See *Emulating Real-Life Conditions*.

For information on using these settings, see *Load Testing - Common Tasks*.

5. Once you are done with the test, select **File | Save** from LoadComplete's main menu to save it.

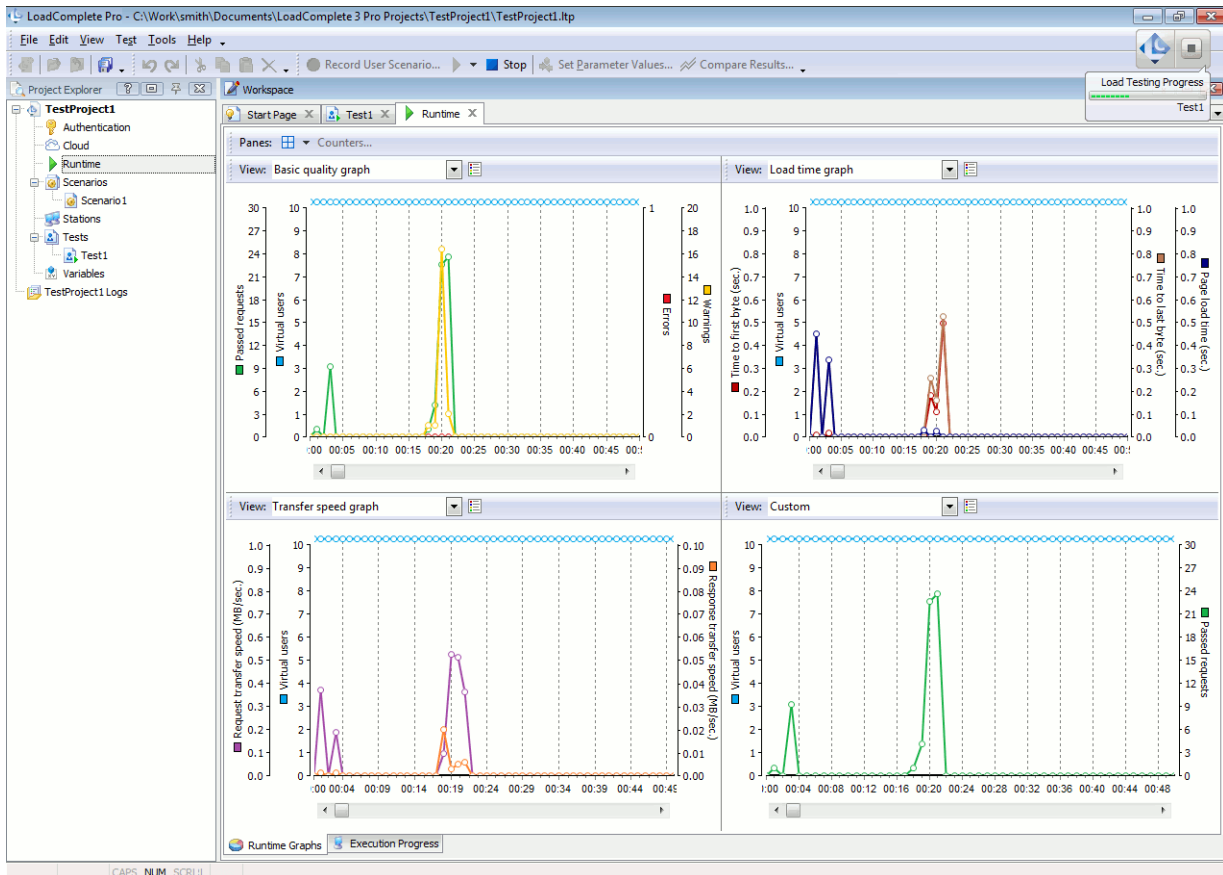
6. Running Your Load Test

After you are done with your load test, you can run it. To do this, click  **Run This Load Test** from the test editor's toolbar. LoadComplete will start the test run.



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

To monitor the test execution, use the *Runtime Graphs* tabbed page of the *Runtime* panel. This page provides you with information on the number of simulated virtual users, the number of errors and warnings that occurred during the run. Each metric is shown in a certain color. For example, the number of simulated users is shown in blue. Here is a sample view of the page:



The Runtime Graphs page

Note that this panel stores statistics only for the last run. You can obtain this graph not only during the test run, but also after the execution is over. However, as soon as you run the test again, the previous results are lost.

After the test run is over, LoadComplete will display test results on screen. For information on analyzing test results, see the *Analyzing Results* section of this tutorial.

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.

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		25	
	C		
Creating		10	
Load tests		21	
Test projects		10	
Creating load tests		21	
	L		
Load testing			
Basic concepts		4	
LoadComplete user interface overview		7	
	R		
Recording			
Scenarios		11	
Recording user scenarios		11	
Running load tests		24	
	S		
Scenarios			
Recording		11	
Verifying		19	
Viewing		17	
	T		
Tips		5	
Emulating real-life conditions		5	
	V		
Verifying recorded scenarios		19	
Viewing			
Recorded scenario		17	
Viewing the recorded scenario		17	